

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Estrutura de Otimização da Cadeia de Abastecimento Global da *Farfetch*

Diogo João da Costa Fernandes

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Alexandre Rodrigues João

Co-orientador: António Almeida e Pedro Borges

24 de Julho de 2018

Resumo

A *Farfetch*, uma empresa de retalho *online* para a moda de luxo, lida diariamente com milhares de encomendas vindas de todo o mundo. O seu modelo de negócio assenta em três grandes pilares: o seu portal, os seus clientes e os seus parceiros. Uma das suas maiores particularidades é o facto de não possuir qualquer *stock* de artigos que se encontram para venda na plataforma *online Farfetch.com*, o que implica a existência de parceiros espalhados pelo mundo que assegurem a gestão de todos os produtos e o cumprimento dos pedidos diários dos clientes.

Sempre que um novo pedido é confirmado pelo cliente no portal, uma das principais tarefas da empresa, executada através do serviço de roteamento, é iniciada e consiste na escolha do melhor *stock point* para expedição da encomenda, uma etapa chave e ao mesmo tempo crítica na sua estratégia de crescimento que necessita de ser explorada.

O presente projeto visa desenvolver uma estrutura para a cadeia de abastecimento da *Farfetch* capaz de avaliar a alocação de parceiros realizada no passado segundo a definição de uma função objetivo composta por três diferentes critérios: lucro, minimizando os custos por pedido, parceiros, garantindo o seu crescimento e clientes, assegurando a sua satisfação. É importante perceber quais as limitações atuais e oportunidades de melhoria que a empresa pode suportar, quais as variáveis a modelar no âmbito dos três critérios mencionados, como avaliar o impacto que esses critérios trazem e como olhar para o futuro tendo em conta esse impacto.

A estrutura implementada foi concebida como um conjunto de blocos interligados com o intuito de alcançar o melhor *trade-off* entre os critérios. Foram eles: modelação, simulação, avaliação, otimização e validação. Na modelação foram estudadas as variáveis a considerar em cada perspetiva e a forma como se relacionariam. Para a visão de lucro foram analisados custos de transporte, comissões e *geomargin*, para parceiros, as vendas totais semanais e o peso das vendas realizadas a partir do portal e para clientes, o tempo de entrega de uma encomenda. A simulação seguiu a modelação efetuada e acedeu a dados reais a partir da base de dados, mais especificamente ao histórico de encomendas para um dia, para calcular as métricas referidas.

A avaliação foi realizada com base nos resultados da simulação e concretizada em estatísticas de modo a perceber se os valores desejáveis para cada perspetiva foram, ou não, atingidos. Como forma de melhorar a resposta dada pela avaliação, a otimização seguiu uma abordagem muito específica, uma rede neuronal *feedforward* multi-camada, com o intuito de treinar os dados recebidos da simulação por *backpropagation* e, assim, atingir a solução ótima, as ponderações ideais para cada critério. A solução ótima é posta à prova na última etapa, a validação, que volta a simular dados e a verificar se os *targets* em cada critério foram aproximados.

Os resultados da simulação revelaram alguma diferença face aos *targets* estabelecidos, o que era de esperar. Com a aplicação da metodologia de otimização, os resultados tornaram-se mais satisfatórios uma vez que a solução ótima obtida provou aproximar duas das três perspetivas dos respetivos *targets*.

Abstract

Farfetch, an online retailer for luxury fashion, deals daily with thousands of orders from around the world. Its business model is based on three main pillars: its portal, its customers, and its partners. One of its major particularities is the fact that it does not have stock, which implies the existence of partners around the world to assure the management of all the products and the fulfillment of daily requests.

Whenever a client places an order in the portal, one of the main tasks of the company, executed through the routing service, consists on choosing the best stock point for shipping the order, a key step and, at the same time, critical in its growth strategy that needs to be explored.

The current project aims to develop a framework for Farfetch's supply chain capable of evaluating the allocation of partners made in the past according to the definition of an objective function composed of three different criteria: profit, minimizing costs per order, partners, ensuring their growth, and customers, ensuring their satisfaction. It is important to understand the current limitations and opportunities for improvement that the company can support, which are the variables to model in the context of the three perspectives mentioned, how to evaluate the impact that these criteria bring and how to look to the future taking into account this impact.

The implemented framework was conceived as a set of interconnected blocks in order to achieve the best trade-off between the criteria. They were: modeling, simulation, evaluation, optimization and validation. In the modeling was studied the variables considered in each perspective and how they would relate. The implemented framework was conceived as a set of interconnected blocks in order to achieve the best trade-off between the perspectives. For the profit vision were analyzed shipping costs, commissions, and geomargin, in the partners' perspective were studied the total weekly sales and the weight of sales made from the portal, and for customers the variable analyzed was the delivery time of an order. The simulation followed the modeling and accesses real data from the database, more specifically to the order history for a day, to calculate the metrics referred.

The evaluation performed is based on simulation results and materialized on statistics in order to see if the desired values for each perspective were or not reached. As a way to improve the response given by the evaluation, the optimization followed a very specific approach, a multilayer feedforward neural network, to train the data received from the simulation by backpropagation and, thus, to reach the optimum solution, the ideal weights for each criteria. The optimum solution is tested in the last step, the validation, which simulates data again and to verify that the targets in each criteria have been approximated.

The simulation results revealed some differences from the established targets, which was expected. With the application of the optimization methodology, the results became more satisfactory since the optimal solution obtained approximated two of the three perspectives to the respective targets.

Agradecimentos

Dizer obrigado, às vezes, não é suficiente para agradecer gestos de pessoas que, nos momentos mais marcantes da nossa vida, estiveram sempre presentes e nos ajudaram da forma que melhor sabiam. O finalizar de um curso de 5 anos é, sem dúvida, um desses momentos e que merece o meu profundo agradecimento às pessoas que acompanharam de perto este meu percurso.

Em primeiro lugar, e porque sem eles eu não teria alcançado esta etapa, agradeço aos meus pais pelo grande apoio nos bons e maus momentos, incentivo diário, compreensão e paciência manifestados ao longo da minha caminhada enquanto estudante. A toda a minha família pela constante preocupação e transmissão de boas energias.

A todos os meus companheiros que percorreram este caminho comigo, obrigado pela vossa dedicação em cada projeto realizado, pela entre-ajuda, pela motivação e pelos momentos divertidos que sempre faziam falta.

Um agradecimento especial à Filipa Nunes, parceira de estágio nestes últimos meses na *Farfetch*, pela companhia, pela simpatia, pela ajuda, pela partilha e por me fazer ver sempre o lado positivo da vida, mesmo quando nem tudo corria bem.

Ao Engenheiro Pedro João, o orientador desta dissertação, pelos conselhos, preocupação e disponibilidade demonstradas ao longo da realização deste projeto.

Gostaria de agradecer à *Farfetch* pela fantástica oportunidade que me foi dada em realizar este projeto e a todas as pessoas com quem tive o privilégio de me cruzar dia após dia e que se revelaram essenciais, quer para a integração na empresa, como para o desenvolvimento do projeto. É inevitável não destacar algumas pessoas. Sílvia Rocha, um obrigado pela simpatia, carinho e, sobretudo, pela preocupação acerca da evolução do projeto. Pedro Borges, obrigado pelo interesse demonstrado no projeto, pelas conselhos e opiniões sempre importantes. E ao António Almeida, um gigante obrigado pelo acompanhamento permanente, pelas ideias transmitidas, pelo constante incentivo e *feedback* e pela simpatia. Foi uma enorme prazer ter-vos conhecido e ter trabalhado convosco. Obrigado por tudo.

Diogo João da Costa Fernandes

*“Pedras no caminho? Eu guardo todas.
Um dia vou construir um castelo.”*

Fernando Pessoa

Conteúdo

1	Introdução	1
1.1	Apresentação da Empresa	1
1.2	Motivação e Objetivos do Projeto	3
1.3	Metodologia	4
1.4	Estrutura da Dissertação	4
2	Estado da Arte	7
2.1	Introdução	7
2.2	Logística	7
2.3	<i>E-commerce</i> e Cadeia de Abastecimento	8
2.4	Otimização da Cadeia de Abastecimento	10
2.4.1	Programação Linear e não Linear Inteira Mista	10
2.4.2	Metaheurísticas	11
2.4.3	Abordagem <i>Fuzzy</i>	14
2.4.4	Redes Neurais Artificiais	17
2.5	Roteamento 4PL	19
2.5.1	Modelo de Programação Incerto (UPM)	20
2.5.2	Modelo de Programação Estocástico (SPM)	21
2.6	Conclusões	21
3	Situação Atual da <i>Farfetch</i>	23
3.1	Posicionamento da <i>Farfetch</i>	23
3.2	Parceiros	24
3.2.1	Operações dos Parceiros	24
3.2.2	Processamento de um Pedido	26
3.3	<i>Stock Points</i>	27
3.3.1	Rotas	27
3.4	Transportadoras	27
3.5	Relação com o Cliente	28
3.6	Gestão da Cadeia	28
3.6.1	Visão de Produto	29
3.7	Limitações do Serviço de Roteamento	31
4	Solução Proposta	33
4.1	Visão Geral da Estrutura de Otimização	33
4.2	Modelação	34
4.2.1	Lucro	35
4.2.2	Parceiros	38

4.2.3	Clientes	39
4.3	Definição da Hipótese	40
4.4	Recolha de Dados	41
4.5	Simulação	42
4.5.1	Etapas Iniciais	43
4.5.2	Lógica	45
4.5.3	Resultados	48
4.6	Otimização	49
4.6.1	Inicialização	50
4.6.2	Propagação Direta	51
4.6.3	Retropropagação	53
4.6.4	Treino	55
4.6.5	Resultados	57
4.7	Validação da Hipótese	58
5	Conclusões e Trabalho Futuro	59
5.1	Conclusão Final	59
5.2	Trabalho Futuro	61
	Referências	63
	Anexo A: Diagrama de Gantt	65
	Anexo B: <i>Queries</i>	67
	Anexo C: Dados Externos	79
	Anexo D: Código da Simulação	81
	Anexo E: Resultados da Simulação	91
	Anexo F: Estatísticas	95
	Anexo G: Rede Neuronal	101
	Anexo H: Resultados da Rede Neuronal	107
	Anexo I: Resultados Após Validação	109

Lista de Figuras

2.1	Camadas e serviços logísticos, em [23]	8
2.2	Processo de fluxo de gestão da cadeia de abastecimento em <i>Chin-Nung Liao et al.</i> , « <i>An integrated fuzzy TOPSIS and MCGP approach to supplier selection in supply chain management</i> », [3], 2011	16
2.3	Rede neuronal <i>feedforward</i> multi-camada em <i>P.C.L. Hui et al.</i> , [21], 2016	19
2.4	Multigrafo para problema de otimização de roteamento em <i>Min Huang et al.</i> , « <i>4PL routing optimization under emergency conditions</i> », [1], 2015	20
3.1	Fluxo tradicional da cadeia de abastecimento	24
3.2	Representação do modelo de negócio da <i>Farfetch</i>	25
3.3	Fluxo principal de um pedido na cadeia de abastecimento da <i>Farfetch</i>	27
4.1	Visão geral da estrutura de otimização desenvolvida	34
4.2	Cálculo do peso volumétrico	37
4.3	Configuração da rede neuronal projetada	51
4.4	Função sigmoid	52
4.5	Algoritmo do processo de treino	57

Lista de Tabelas

1.1 Diagrama de tarefas 5

Abreviaturas e Símbolos

1PL	Primeiro grau de terceirização de atividades logísticas
2PL	Segundo grau de terceirização de atividades logísticas
3PL	Logística subcontratada
4PL	Logística quarterizada
4PLROP	Problema de otimização de roteamento na quarta camada logística
AIMMS	Sistema de modelação multidimensional interativo avançado
AHP	Processo de hierarquia analítica
ANN	Rede neuronal artificial
B2B	Negócio para negócio
B2C	Negócio para cliente
bw	Largura de banda
C2C	Cliente para cliente
DCC	Centro de distribuição
DDP	Taxas de entrega pagas
DDU	Taxas de entrega não pagas
DM	Decisores
FMOM	Modelo difuso multi-objetivo
FNIS	Solução ideal difusa negativa
FPIS	Solução ideal difusa positiva
HMCR	Taxa de consideração de memória de harmonia
HMS	Tamanho da memória de harmonia
HS	Pesquisa de harmonia
KBGA	Algoritmo genético baseado no conhecimento
KBC	Cruzamento baseado no conhecimento
KBI	Inicialização baseada no conhecimento
KBS	Seleção baseada no conhecimento
LINGO	<i>Software</i> de modelação para programação linear, não linear e inteira
MCAL	Níveis de aspiração de múltipla escolha

MCDM	Tomada de decisão de múltiplos critérios
MCGP	Programação de metas de múltipla escolha
MILP	Programação linear inteira mista
MOP	Programação multi-objetivo
NPS	Pontuação líquida do promotor
OMS	Serviço de gestão de encomendas
PAR	Taxa de ajuste de inclinação
TOPSIS	Técnicas para preferência de pedidos por semelhança com a solução ideal
SC	Cadeia de abastecimento
SCN	Rede da cadeia de abastecimento
SGA	Algoritmo genético simples
SPM	Modelo de programação estocástico
STORM	Gestão de ordens de inventário
TPV	Valor total de aquisição
TSP	Problema de vendedor ambulante
UPM	Modelo de programação incerto

Capítulo 1

Introdução

A presente dissertação foi desenvolvida no âmbito do Mestrado Integrado em Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto.

O projeto descrito foi desenvolvido em ambiente empresarial, na *Farfetch*, em Leça do Balio, no período de cerca de 4 meses e meio, com início a 5 de Fevereiro de 2018 e fim a 15 de Junho de 2018, com supervisão do Eng^o António Almeida, atual *Senior Product Owner*.

1.1 Apresentação da Empresa

A *Farfetch* é uma plataforma *e-commerce* para a moda de luxo que surge do espírito visionário de um empreendedor português, José Neves, que tinha duas empresas de setores distintos: uma de programação e outra de calçado. Desde cedo se dedica ao mundo da moda, desde a década de 90, quando lançou a marca de sapatos *SWEAR*. Mais tarde, criou a *B Store*, uma empresa de moda com uma loja física com marcas jovens e *designers* emergentes. Foi durante a promoção da *B Store* na semana da moda em Paris, em Outubro de 2007, que tudo começou.

Já com várias *startups* no seu currículo, José Neves percebeu que não conseguia vender os seus produtos porque as vendas nas lojas físicas estavam a estagnar ao passo que o comércio *online* disparava. Foi nesse momento que surgiu a oportunidade de criar um *marketplace*, uma junção de lojas *online*, que lhe permitia ter um negócio à escala mundial ao mesmo tempo que ajudava as lojas com quem trabalhava a vender mais. A ideia era permitir que pequenas lojas independentes pudessem competir no mercado mantendo as suas lojas físicas, mas criando uma identidade própria.

Um ano depois, em 2008, nasce a *Farfetch*. O conceito era aparentemente simples: tratava-se de criar uma ponte *online* entre consumidores e lojas, independentemente da sua localização. Por exemplo, um cliente em Lisboa poderia comprar uma peça de luxo em Hong Kong e receber a sua encomenda em casa. A *Farfetch* trataria de todo o processo.

No período de 2010 a 2017 foram oito, no total, as rondas de investimento na *Farfetch*. Em Julho de 2010, foi realizado um investimento de \$4.5M pela *Advent Venture Partners*. Cerca de dois anos depois, em Janeiro de 2012, foi a vez de três outros investidores dirigidos pela *Index*

Ventures investirem cerca de \$18M. Menos de 1 ano depois, em Março de 2013, o número de investidores crescia, bem como o valor do investimento realizado. Nesse ano, foram quatro os investidores, entre eles a *Condé Nast*, num valor total de \$20M. No ano seguinte, em Maio, a *Vitruvian Partners* juntamente com mais 7 investidores decidiram também apostar na *Farfetch*, investindo \$66M.

A empresa continuava a crescer, os investimentos a surgir e o ano de 2015 não foi exceção, com um investimento total de \$86M realizado por 8 investidores liderados pela *DST Global*. 2016 foi também um ano de bastante investimento nesta plataforma, com a *Eurazeo* e a *IDG Capital Partner* em conjunto com mais 6 investidores a aplicar um financiamento de \$110M. Mas como o melhor fica sempre guardado para o fim, 2017 viria a bater todos os recordes ao registar um investimento de \$397M vindo de um único investidor, a *JD.com*.

No total, foram já investidos \$721.5M nesta empresa no período de 7 anos.

Atualmente, a *Farfetch* envia produtos de moda de mais de 750 *designers* e lojas parceiras da plataforma para mais de 170 países. As maiores vendas da *Farfetch* vêm de marcas de luxo como a *Valentino*, *Saint Laurent*, *Givenchy* e *Gucci*, mas *designers* emergentes e menos conhecidos são a chave diferenciadora da empresa.

Além do portal, a *Farfetch* disponibiliza também uma *app* compatível com *android* e *iphone* que permite aos consumidores comprarem peças de luxo através do seu *smartphone*, bem como aceder a guias de viagens para algumas das cidades mais cosmopolitas do mundo, como Nova Iorque, Paris e Los Angeles.

A plataforma de comércio *online* tem escritórios em Londres, Porto, Guimarães, Lisboa, Nova Iorque, Los Angeles, São Paulo, Tóquio, Hong Kong, Shanghai e Moscovo, com uma equipa de quase 1500 colaboradores. Embora a sede se localize em Londres, é em Leça do Balio, na Lionesa, que está toda a equipa de desenvolvimento, serviços financeiros, bem como a parte de atendimento ao cliente. Todas as peças disponíveis na plataforma são fotografadas e catalogadas noutro edifício da empresa situado em Guimarães.

No que ao escritório do Porto diz respeito, está atualmente dividido em onze departamentos principais: Operações, Gestão de conta, Tecnologia, Finanças, Serviço ao cliente, Serviço ao parceiro, Recursos humanos e Produto.

No campo da inovação e da consolidação da sua estratégia de crescimento, a empresa lançou e implementou a ideia denominada de *Store too Door in 90 minutes* que permite comprar peças da *Gucci* através da plataforma *FF.com* e também a partir da aplicação móvel, assim como receber esses pedidos em casa em apenas 90 minutos, provenientes de lojas da marca em Londres, Nova Iorque, Dubai, Los Angeles, Madrid, Milão, Paris, São Paulo, Miami e Tóquio.

Neste seguimento, prepara-se para lançar a *Store of the future*, uma loja física focada na personalização da compra com base na recolha de dados sobre os clientes, uma mistura de tecnologia e experiência em loja. Londres e Nova Iorque serão os pioneiros no sentido em que a tecnologia utilizada na *Store of the future* será implementada na loja da *Browns*, em Londres e na *Thom Browne*, em Nova Iorque.

A empresa segue valores que considera essenciais para o seu espírito inovador: ser humano, ser brilhante, todos juntos, ser revolucionário, pensamento global e surpreender os clientes no dia a dia.

1.2 Motivação e Objetivos do Projeto

A *Farfetch* gere cerca de 6000 pedidos por dia, com parceiros e clientes distribuídos por todo o mundo. Para um determinado *item* comprado *online*, pode existir mais do que um fornecedor capaz de atender esse pedido, localizado numa parte exclusiva do mundo com determinadas especificações. Um mesmo artigo pode ter diferentes preços de venda em diferentes lojas, sendo esta oscilação de preços entre parceiros um fator de extrema importância no momento de decidir um parceiro em detrimento de outro.

Portanto, identificar o fornecedor adequado, de entre um conjunto de parceiros, capaz de cumprir cada encomenda *online* é considerada uma tarefa crítica, chave e necessária a ser executada para o crescimento sustentável da empresa com princípios bem definidos no âmbito de três importantes perspectivas:

- **Lucro:** Minimização dos custos, levando em consideração o valor e todos os custos diretos e indiretos envolvidos na satisfação da encomenda;
- **Parceiros:** Garantir um sistema democrático para alocação de ordens, capaz de apoiar o crescimento dos parceiros;
- **Cliente:** Maximização dos níveis de atendimento.

Sendo a *Farfetch* uma plataforma que se limita a gerir a logística de venda *online*, não tendo qualquer *stock* mas sim múltiplos pontos de *stock* pelo mundo para expedição de produtos, é natural que possua custos e preços de transporte mais elevados quando comparados com os dos seus concorrentes de negócio que têm armazéns e *stock* para puderem expedir de imediato as encomendas.

Nesse sentido, os principais objetivos para este projeto são os seguintes:

- Revisão de literatura para compreensão e avaliação de assuntos relevantes em torno do projeto: análise de estudos realizados em cadeias de abastecimento inseridas em diversas indústrias no que diz respeito a métodos e modelos implementados para a sua otimização;
- Analisar e compreender benefícios e limitações das metodologias existentes na *Farfetch* no âmbito de produto para alocação de parceiros: identificar oportunidades de melhoria;
- Projetar uma estrutura capaz de avaliar alocações passadas de parceiros de acordo com a definição de uma função objetivo que englobe as três perspectivas delineadas, assegurando flexibilidade, confiabilidade e rastreabilidade;
- Desenvolver um protótipo para avaliação do modelo: Análise *As-Is* vs *To-Be*.

Trata-se de um problema de otimização multi-variável e multi-objetivo para maximização da performance logística da *Farfetch* e diminuição do tempo de colocação em *live*.

1.3 Metodologia

Considerando os objetivos expostos e delineados na subsecção 1.2, a metodologia implementada na abordagem ao projeto foi rigorosamente planeada através de um conjunto de tarefas e respetiva previsão para a sua duração, tendo em conta o tempo total disponível, como mostra a tabela 1.1. Sempre que uma tarefa era iniciada e terminada, as datas respetivas eram registadas, assim como se o prazo para a sua execução foi, ou não, cumprido. A evolução das tarefas necessárias à realização do projeto pode, também, ser visualizada graficamente através de um diagrama de *gantt*, exposto no anexo A.

A tarefa inicial, a realização de um programa de indução, ocorreu nas duas primeiras semanas e consistiu em ganhar conhecimento acerca do modelo de negócio da empresa, conversando e reunindo com diversas pessoas de diferentes departamentos, percebendo como interagem e quais as suas funções, facilitando assim a integração na empresa.

Seguiu-se o estado da arte, uma tarefa de pesquisa e estudo acerca daquilo que já existe no mercado no âmbito do projeto a desenvolver e que ajuda na seleção da melhor abordagem a seguir.

Realizadas as duas primeiras tarefas, o foco passa a incidir totalmente no projeto, começando pela identificação do problema e definição dos objetivos a atingir, para depois se avançar para a próxima tarefa, perceber em que ponto se encontra a empresa no contexto em que vamos trabalhar.

As quatro tarefas seguintes são as mais complexas e que requerem mais tempo, inserindo-se na estrutura a desenvolver de acordo com a estratégia da *Farfetch*, recorrendo-se para isso a um editor de texto, o *Atom* e à linguagem *python*.

As última tarefa, a escrita da dissertação, ocorreu simultaneamente com o decorrer das tarefas mencionadas anteriormente.

1.4 Estrutura da Dissertação

Para além da introdução, a presente dissertação está organizada em quatro capítulos distintos.

No capítulo 2 é descrito o estado da arte, começando pela apresentação de alguns conceitos úteis e importantes no âmbito de qualquer cadeia de abastecimento, para depois ser abordada a questão da otimização, o foco principal desta dissertação, referindo e revendo várias metodologias aplicadas em diferentes contextos consoante os objetivos pretendidos.

No capítulo 3 é apresentada uma visão geral do modelo de negócio da *Farfetch*, incidindo sobre as entidades cruciais envolvidas na sua cadeia de abastecimento. Posto isto, entra-se no domínio de produto, nomeadamente na forma como está articulado e como funciona ao nível do processamento de um pedido, com destaque para a seleção de parceiros na satisfação de uma encomenda. As limitações do seu funcionamento atual são também abordadas.

Tabela 1.1: Diagrama de tarefas

Tarefa	Duração prevista (dias)	Data de início	Data de término	Realizada na duração prevista?
Programa de indução	10	5 de Fevereiro	16 de Fevereiro	Sim
Estado da arte	20	12 de Fevereiro	2 de Março	Sim
Definição dos objetivos do projeto	5	19 de Fevereiro	23 de Fevereiro	Sim
Situação atual da empresa no âmbito do projeto	10	26 de Fevereiro	9 de Março	Sim
Modelação do problema	10	12 de Março	23 de Março	Sim
Recolha e tratamento de dados	5	26 de Março	30 de Março	Sim
Realização da simulação	20	2 de Abril	30 de Abril	Sim
Método Preditivo	25	30 de Abril	1 de Junho	Sim
Escrita da dissertação	25	12 de Fevereiro	11 de Junho	Sim

No capítulo 4 é descrita a solução proposta para o problema inicialmente identificado e está dividido em 5 grandes etapas: modelação, simulação, avaliação, otimização e validação. Na modelação são explicadas as métricas consideradas e a forma como se relacionam, para de seguida serem simuladas recorrendo ao histórico de encomendas. A avaliação compara resultados que depois são utilizados no método preditivo. Por fim, e utilizando a solução ótima obtida, é simulada a hipótese definida para comprovar melhorias nos resultados.

No capítulo 5 são feitas algumas considerações e realces acerca do projeto, são discutidos os resultados obtidos e são sugeridos alguns avanços e trabalhos futuros consoante o que foi surgindo ao longo do projeto.

Capítulo 2

Estado da Arte

2.1 Introdução

Neste capítulo é realizada uma revisão bibliográfica de assuntos relevantes como base de sustentação à implementação do projeto. Inicialmente, na secção 2.2, são apresentados os serviços e/ou camadas logísticas e a forma como se integram na cadeia de abastecimento. Seguidamente, na secção 2.3, evidencia-se o conceito de *e-commerce* e a sua relação com a cadeia de abastecimento. Abordados os conceitos gerais no âmbito da cadeia, direciona-se a atenção na secção 2.4 e 2.5 para a sua otimização no que diz respeito a métodos e estratégias adotadas em casos práticos.

2.2 Logística

Os serviços de logística, responsáveis por garantir a execução de todas as atividades de uma organização através da promoção de recursos e informações, podem ser classificados em quatro camadas inter-relacionadas: 1PL, 2PL, 3PL e 4PL, tal como evidenciado na Figura 2.1.

As empresas são classificadas como *first-party logistics* quando se mostram capazes de gerir as suas atividades sem a intervenção de terceiros dada a sua baixa complexidade. Quando a empresa inicia um processo de expansão, surge o conceito de *second-party logistics* que envolve a terceirização de atividades logísticas que antes eram executadas dentro da empresa.

O conceito *third-party logistics* refere-se a quem efetivamente realiza as operações logísticas da empresa. Pode atuar em diferentes domínios, desde o transporte e embalagem até ao armazenamento e controlo de *stock*. Além disso, o 3PL pode igualmente seguir uma estratégia de customização de acordo com as necessidades específicas de cada cliente.

Na base da pirâmide está o *fourth-party logistics*, responsável por gerir e coordenar o 3PL, um integrador da cadeia de abastecimento que desenvolve uma atividade ampla em torno da sua gestão e não uma operação em particular como acontece com os outros serviços do setor logístico empresarial. Este serviço acompanha todos os processos, reúne e gere recursos, capacidades e tecnologia da sua própria organização em conjunto com serviços complementares prestados com

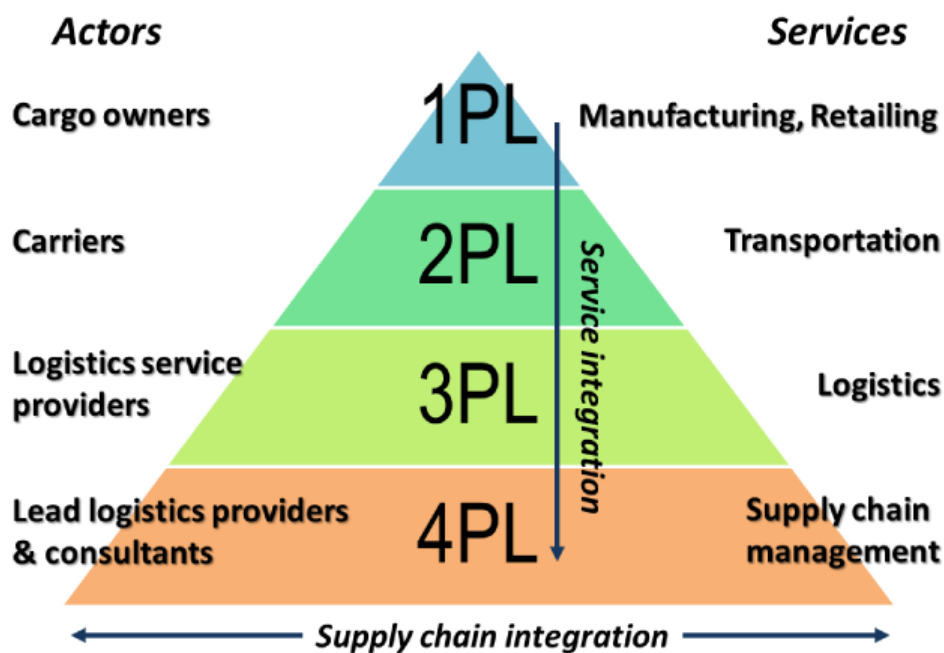


Figura 2.1: Camadas e serviços logísticos, em [23]

vista à obtenção de uma solução abrangente para a cadeia de abastecimento. Tal meta implica uma comunicação constante entre todos os membros da cadeia.

A aplicação deste serviço, nos dias de hoje, não é muito comum entre as pequenas e médias empresas ao contrário do que acontece nas grandes empresas que vêm apostando no desenvolvimento deste conceito que, como qualquer outro, traz vantagens e desvantagens associadas.

2.3 E-commerce e Cadeia de Abastecimento

São cada vez mais e maiores os desafios que as organizações empresariais enfrentam no seu dia a dia que estimulam e afetam a sua definição operacional e estratégica, levando a conflitos naturais na organização, nos parceiros e nos clientes.

O *e-commerce*, comércio eletrônico, trata-se de um conceito de negócio que se caracteriza pela compra e venda de bens ou serviços eletronicamente e que inclui três vertentes distintas:

- **Business to Consumer (B2C):** O *website* é o local onde ocorrem diretamente as transações entre uma organização empresarial e o consumidor. A organização do negócio após receber uma encomenda, enviará os bens para o cliente. Alguns exemplos de sucesso são a *Amazon.com* e a *Priceline.com* (Yu *et al.*, 2016, [6]).
- **Business to Business (B2B):** Comércio de bens e serviços entre organizações. O volume de transações em B2B é maior do que o volume em B2C (Yu *et al.*, 2016, [6]).

- **Costumer to Costumer (C2C):** A empresa apenas fornece a plataforma para que as pessoas a utilizem e possam interagir com benefício mútuo. É o exemplo da Uber ou da OLX.

A transferência e gestão de informação de qualquer processo e entre processos é feita igualmente de forma eletrónica através da plataforma.

Trata-se de um tipo de negócio que está, claramente, a proliferar nas empresas através de novos canais de venda, aceleração da internacionalização e novos competidores até porque, quando bem implementado, traz vantagens significativas quando comparado com os métodos tradicionais de transferência de bens e serviços:

- **Mais barato:** Desenvolver uma loja *online* tem custos mais reduzidos do que uma loja física;
- **Mais rápido:** Existem plataformas com todas as ferramentas já implementadas prontas a serem operacionalizadas;
- **Disponibilidade:** A loja *online* está disponível 24h por dia durante 7 dias por semana;
- **Acessibilidade:** Qualquer pessoa em qualquer parte do mundo pode aceder;

A crescente capacidade associada ao forte investimento em tecnologia e comunicação faz do comércio eletrónico uma atividade com um impacto relevante na cadeia de abastecimento ao nível dos seus três principais fluxos: físico, financeiro e informação.

Os negócios de comércio eletrónico utilizam muito os atores da camada 3PL, os *logistics service providers*, evidenciados na Figura 2.1, para supervisionar e gerir a cadeia de abastecimento. Neste âmbito, especializam-se na sua otimização, permitindo que as lojas *online* se concentrem noutras operações do negócio.

As empresas devem procurar, crescentemente, uma orientação totalmente voltada para o cliente, baseando-se em conhecimento e informação, investindo vigorosamente em processos de parceria, embora não seja um passo fácil.

Uma comunicação eficaz entre todos os membros da cadeia de abastecimento integrada, já mencionada na secção 2.1, é reconhecidamente um fator de competitividade para as organizações no sentido em que permite a ligação e interação entre as várias atividades logísticas incorporadas num processo integrado.

A cadeia de abastecimento integrada está difundida a um conjunto de variáveis internas e externas que afetam a organização e os diferentes modelos de negócio subjacentes. Antes de procurar melhorar e otimizar a cadeia de abastecimento, é imprescindível conhecer e empresa no seu todo em termos de organização e funcionamento no que ao processo de fornecimento diz respeito.

O conceito de cadeia de abastecimento integrada surgiu da evolução do conceito de cadeia logística, trazendo consigo uma visão mais ampla do negócio e melhorando conceitos como movimentação de materiais e distribuição física de produtos, mais limitados na cadeia logística.

2.4 Otimização da Cadeia de Abastecimento

Existem diferentes e diversos estudos e abordagens que lidam com a otimização da cadeia de abastecimento onde o principal foco é a minimização de custos ou maximização de lucro.

Quando referimos o conceito de otimização da cadeia de abastecimento, é importante perceber que existem dois tipos de otimização: otimização local e otimização global.

Sendo a cadeia de abastecimento uma interação complexa entre diferentes missões, por otimizar entenda-se equilibrar a cadeia em termos de eficiência e capacidade de resposta ao menor custo total.

A otimização local foca-se numa função em específico sem considerar a forma como esta afeta toda a cadeia. Este método tenta encontrar um mínimo local sem garantia que o mínimo global seja encontrado. Para a maioria das situações, a utilização de um algoritmo de otimização local não leva à melhor resposta.

Com a otimização global aquilo que se procura é uma verticalidade em toda a cadeia. Os custos podem aumentar para alguns dos membros em particular mas para a cadeia no geral esses custos são reduzidos. Todas as entidades colaboram na troca de informação e ideias entre parceiros, sendo este trabalho conjunto que permite chegar a um saldo final positivo.

2.4.1 Programação Linear e não Linear Inteira Mista

Liu, Lin e Yeh (2017, [20]) implementou um sistema metódico e eficiente para a gestão da cadeia de abastecimento baseado em dois métodos de categorização. O primeiro método envolve dividir a cadeia nas seguintes áreas consoante os seus níveis de decisão:

- Áreas estratégicas: decisões a longo prazo;
- Áreas táticas: decisões a médio prazo;
- Áreas operacionais: decisões a curto prazo.

O segundo método de categorização divide a cadeia em decisões *upstream*, *midstream* e *downstream* consoante os seus procedimentos operacionais.

Vários estudos foram realizados e no seu seguimento vários modelos de otimização implementados para diferentes tipos de processos e focados nas várias categorias acima referidas.

Modelos de programação não linear inteira mista com vista à maximização de lucro e à resolução de problemas envolvendo decisões táticas *upstream* e *midstream*. As variáveis de decisão consideradas foram de médio prazo, incluindo o número de matérias-primas adquiridas ao fornecedor, número de matérias-primas armazenadas e número de matérias-primas processadas. Estes modelos foram, posteriormente, extendidos a **modelos de programação estocásticos** para lidar com a incerteza da oferta.

Para questões estratégicas, como a localização de instalações ou centros de distribuição, foram desenvolvidos **modelos de programação linear inteira mista**. A localização das instalações, as

capacidades de produção e as distâncias de transporte anuais foram designadas como variáveis de decisão. O objetivo do modelo passou pela minimização dos custos para as organizações.

Os modelos de otimização revelaram-se escaláveis e flexíveis a novas variáveis de decisão *downstream*, como a procura e/ou vendas, sem que as operações *upstream* e *midstream* fossem alteradas. Deste modo, a otimização compreendia todos os setores da cadeia: matérias-primas, fornecedores, transporte, armazenamento, produção e procura.

O *software* de modelação multidimensional integrado utilizado (AIMMS) permite a construção do modelo de programação matemático, a realização de simulações, análise de cenários e sensibilidade.

Os resultados das simulações mostravam mudanças no inventário diário e nos custos finais da rede da cadeia de abastecimento. A análise de sensibilidade permitia perceber quais os parâmetros mais críticos para a maximização do lucro para, desse modo, serem intervencionados.

As vantagens de lucro alcançadas com a implementação desta estratégia revelaram-se consideráveis. No entanto, quando ocorriam incertezas, o modelo poderia manter ou perder as suas vantagens, verificando-se ser esta a sua principal limitação.

Também Bittante, Pettersson e Sax En (2018, [18]) desenvolveu um modelo matemático mas desta feita com o intuito de resolver um problema estratégico de transporte marítimo.

A formulação do problema considerou um conjunto de localizações de portos divididos em dois subconjuntos: um primeiro subconjunto de portos de abastecimento (fornecedores) e um segundo subconjunto de portos de receção (clientes). Diferentes tipos de navios para transporte foram também considerados: um subconjunto de navios disponíveis para transporte e um outro em que não estão aptos para o transporte de carga.

O roteamento dos navios, os custos de navegação e os custos de fretamento foram modelados usando equações lineares e restrições através de um **modelo de programação linear inteira mista**. A função objetivo caracterizava-se pela minimização do total dos custos combinados: custos de transporte, custos de aluguer dos navios e custos do produto a ser transportado.

A performance do modelo foi ilustrada através de um caso base cuja solução obtida foi a desejada. Dessa forma, o problema foi expandido para uma análise de sensibilidade com o intuito de estudar a evolução da solução ideal perante outras condições e potencial do modelo.

Para sistemas mais complexos, com dezenas de portos, o modelo comportou-se adequadamente, mostrando a sua eficiência. Mas também que o tempo computacional aumenta com o aumento da complexidade do problema.

2.4.2 Metaheurísticas

Bányai, Veres e Illés (2015, [10]) considerou o problema das empresas de manutenção em rede e dos processos de compra de peças de reposição com vista à maximização da utilização de recursos e consequente eficiência de todo o sistema. A questão foi analisada do ponto de vista da otimização da cadeia de abastecimento com foco naquilo a que se designa por ***make-or-buy decision***, ou seja, escolher entre o fabrico da peça internamente ou a sua compra a fornecedores externos.

As peças exigidas pelos processos de manutenção representavam uma parte importante dos custos logísticos e, como tal, era importante otimizar a cadeia do ponto de vista da *make-or-buy decision* para diminuir custos de produção e compra.

O problema descrito foi formulado através de um modelo matemático com equações, restrições e objetivos. O modelo procurou a proporção ideal de produções internas e terceirizadas, bem como a quantidade da ordem, de modo a minimizar os custos.

O modelo foi resolvido recorrendo a metaheurísticas, mais especificamente através de um **algoritmo de pesquisa de harmonia** que busca a solução ótima de *make-or-buy* neste tipo de problemas. Consiste em 5 etapas principais:

1. Definir o problema e colocar os parâmetros do algoritmo;
2. Criar e preencher a matriz de memória de harmonia;
3. Criar ou improvisar uma nova solução;
4. Se a nova solução é melhor do que a pior solução na memória de harmonia, então deve ser substituída;
5. Repetir o terceiro e quarto passo até obter a harmonia perfeita, o denominado ótimo global.

Neste algoritmo existem três técnicas principais para improvisar soluções. A decisão sobre qual técnica usar é auxiliada pela *Harmony Memory Considering Rate* (HMCR) e pela *Pitch Adjustment Rate* (PAR). Existem ainda dois valores importantes a ter em conta: o *Harmony Memory Size* (HMS) que decide quantas soluções se mantêm e a largura de banda (bw) que controla a frequência de modificação de uma variável. As três técnicas são as seguintes:

1. Criar uma solução totalmente aleatória, tal como se preenche a matriz HS;
2. Cada variável na nova solução obtém uma escolha aleatória de outra solução;
3. Tal como a técnica 2 mas modifica-se o valor em alguns casos.

Caso os resultados obtidos em termos de variação de custos não fossem os expectáveis, era realizada uma análise de sensibilidade de modo a encontrar os melhores parâmetros do algoritmo para que os resultados do modelo matemático fossem os ideais. Com uma configuração adequada do HMCR, PAR, bw e HMS, o progresso no processo de otimização foi mais rápido e preciso.

Prakash *et al.* (2012, [15]) propôs um outro algoritmo, o chamado **algoritmo genético baseado em conhecimento (KGBA)**, desta feita para otimização de rede da cadeia de abastecimento (SCN) para uma operação eficiente a longo prazo. O algoritmo proposto integra a base de conhecimento para melhoria da qualidade da solução, partindo da geração da população inicial, seleccionando os indivíduos para reprodução e reproduzindo novos indivíduos.

O KGBA surgiu da extensão e melhoria do desempenho do **algoritmo genético simples (SGA)** como uma nova metodologia para lidar com uma variedade complexa de variáveis típicas

de uma cadeia de abastecimento. Para atingir tal objetivo foram introduzidos novos operadores genéticos: inicialização, seleção, cruzamento e mutação.

Com a aplicação deste algoritmo pretendia-se encontrar um ótimo ideal perante dois objetivos que foram definidos e cuja natureza era conflituosa: minimização da média total de custo para satisfação da procura e maximização da taxa de satisfação da procura.

Tendo como base os critérios de desempenho anteriores, o algoritmo foi aplicado a um caso de estudo de uma rede que envolvia fornecedores externos, um armazém central e retalhistas. O principal problema para otimização da rede residia na avaliação da seleção de diferentes fornecedores localizados em diferentes partes e respetivas rotas de transporte.

Os custos levados em consideração foram diversos: custos de compra, custos de transporte, custos de compromisso, custos de manutenção de inventário, custos de encomenda e custos devido a pedidos não atendidos. Definidos os objetivos, restrições e todos os parâmetros a ter em consideração, o problema foi formulado, uma vez mais, através de um modelo matemático.

No primeiro passo do algoritmo, a **inicialização baseada no conhecimento (KBI)**, uma população inicial de soluções foi gerada, sendo a melhor alternativa guardada na base de conhecimento como ponto de partida para o algoritmo.

Para a inicialização são necessárias várias informações e, com base nessas informações, é formado um cromossoma. Informações sobre o fornecedor, em particular quantos fornecedores são considerados, o custo de compromisso com cada um, a sua localização, os meios de transporte disponíveis e custo associado e também a procura, constituem cada cromossoma.

Na etapa seguinte, a **avaliação**, cada sequência é calculada de acordo com critérios de avaliação (multi-objetivos).

Após a avaliação, segue-se a **seleção com base no conhecimento (KBS)**, onde um subconjunto da população inicial é selecionado de acordo com *Neo Darwinismo* que subdivide o procedimento de seleção em três categorias: seleção direcional, seleção estável e seleção incontrolável. Para executar os três tipos de seleção, existem vários métodos de seleção.

Após a seleção, um subconjunto de cromossomas sobreviventes são selecionados aleatoriamente de acordo com a probabilidade de cruzamento para formarem novos filhos. O **cruzamento baseado no conhecimento (KBC)** fornece as características dos pais aos novos filhos. Tal como acontece na seleção, existem vários esquemas de cruzamento: cruzamento de ponto único, cruzamento de mapeamento parcial, cruzamento uniforme, cruzamento de ciclo, entre outros.

As características de cada esquema de cruzamento bem como o seu desempenho para diferentes ambientes nas várias probabilidades são guardados na base de conhecimento.

No seguimento do passo anterior, vem a **mutação com base no conhecimento**, que em vez de recombinar dois cromossomas, modifica os genes ou bits de um cromossoma único. O desempenho dos vários operadores de mutação (inversão, inserção, deslocamento, entre outros) em diferentes ambientes do sistema com variedade da função objetivo é também armazenado na base de conhecimento.

Após a mutação, as populações selecionadas, de igual tamanho à população inicial, são introduzidas para a próxima geração fora da população dos cromossomas.

O processo é repetido até ser cumprido o critério de terminação definido. Este critério pode ser, por exemplo, número de gerações.

Como resultados da implementação deste algoritmo, de destacar a taxa de convergência, mais rápida no algoritmo KBGA (144 gerações) do que no algoritmo SGA (2000 gerações). A visão baseada no conhecimento revelou-se fundamental para o sucesso do compromisso da satisfação da procura com a média mínima de custo por produto.

Trata-se de uma visão diferente mas com resultados eficientes no âmbito da resolução de problemas combinatórios.

2.4.3 Abordagem *Fuzzy*

Govindan *et al.* (2017, [9]) apresentou um **modelo multi-objetivo difuso (FMOM)** de forma a configurar o projeto de rede de uma cadeia de abastecimento para recuperação de produtos, que integra a seleção de fornecedores, alocação de fluxo e decisões de transporte. A abordagem de programação difusa foi adotada para lidar com a incerteza perante múltiplos objetivos.

O modelo integrado foi desenvolvido de acordo com uma estrutura concetual constituída por três grandes passos:

1. **Processo de avaliação de fornecedores** recorrendo a:

- Identificação de critérios para a seleção de fornecedores;
- Uso da **análise fatorial** para extração de um menor número de fatores interpretáveis significativos;
- Uso da **AHP** para priorizar os fornecedores com base nos fatores.

2. **Técnica *k-means clustering***: A métrica da distância e máxima capacidade do meio de transporte disponível são aplicadas para agrupar centros de distribuição (DCC) em *clusters* ótimos.

3. **Modelação e formulação da rede** nos seguintes objetivos difusos:

- **Minimização do custo líquido da rede**: diferença entre o somatório do custo de processamento, transporte e compra e o lucro total;
- **Maximização da performance dos fornecedores**: seleção de fornecedores com pesos AHP mais altos;
- **Minimização dos efeitos de transporte**: seleção adequada dos meios de transporte para distribuição ao longo das rotas mais curtas determinadas pelo **problema de vendedor ambulante (TSP)**.

A análise fatorial procura criar, num problema com muitas variáveis, um conjunto de novas variáveis que não estão correlacionadas, os chamados fatores. Os fatores têm o objetivo de resumir

as diversas variáveis num conjunto de menores dimensões com perda mínima de informação (Govindan *et al.*, 2017, [9]). A base desta análise está na matriz que representa a correlação entre as variáveis envolvidas e requer uma amostra significativa.

Para lidar com situações subjetivas e conflituosas, o método **AHP** revelou-se o mais indicado e com o principal propósito de priorizar fornecedores com base nos fatores principais e independentes mencionados acima. O método foi aplicado de acordo com os seguintes passos:

1. A matriz de comparação é gerada onde cada elemento (i, j) representa a comparação entre os critérios (i, j) . O juízo subjetivo é quantificado usando uma escala de nove valores, começando pelo 1 ("tão importante como") até ao 9 ("extremamente importante do que").
2. A matriz mencionada no ponto anterior é primeiramente normalizada e, de seguida, as prioridades são calculadas como médias das linhas da matriz normal.
3. Cálculo do índice de consistência e da razão de consistência dos juízos. Uma matriz com uma razão de consistência inferior a 0,1 é aceitável (Govindan *et al.*, 2017, [9]).

A aplicação do algoritmo ***k-means clustering*** permitiu o agrupamento de centros de distribuição em zonas e subzonas de fluxo direto e retorno, considerando a sua proximidade com instalações de montagem e a proximidade entre si. As distâncias foram calculadas recorrendo à norma euclidiana.

O ordenamento dos centros de distribuição ajuda a agilizar o fluxo de transporte onde cada zona receberá meios de transporte para movimento de produtos.

Govindan *et al.* (2017, [9]) comprovou igualmente que num problema de programação multi-objetivo (MOP), otimizar simultaneamente todos os objetivos divergentes, como é o caso, sujeitos às restrições do sistema, dificilmente é possível.

Para garantir flexibilidade às ambições contraditórias dos decisores no que aos objetivos do modelo diz respeito, foi seguida a **abordagem de programação difusa** para ajustar claramente o grau de satisfação de cada objetivo.

O algoritmo para resolução do problema FMOM seguiu os seguintes passos:

1. Determinação dos níveis de aspiração e tolerância para os objetivos difusos, resolvendo os problemas de cada objetivo de forma independente e, em seguida, obter os seus valores otimista e pessimista;
2. Definição das *membership functions* lineares para cada objetivo difuso;
3. Os pesos são atribuídos aos objetivos conforme a preferência dos decisores e representam a sua importância relativa;
4. Conversão do problema inicial num modelo mais simples em termos de tempo computacional e flexibilidade;

A **abordagem ponderada max-min** ajudou a encontrar soluções eficientes para o problema multi-objetivo de modo a que a proporção dos níveis de realização dos objetivos e a proporção dos respetivos pesos se aproximassem o mais possível.

Chen, Lin e Huang (2006, [2]) aplicou o valor linguístico para medir as classificações e os pesos atribuídos aos critérios de seleção de fornecedores, utilizando um **modelo MCDM** baseado na teoria dos conjuntos *fuzzy* num caso prático de cadeia de abastecimento. O modelo revelou-se apenas adequado para problemas *single-sourcing* em que um fornecedor presumivelmente satisfará as necessidades do comprador de tal forma que a decisão deve ser tomada sobre qual será o melhor fornecedor.

No caso de *multi-sourcing*, nenhum fornecedor pode satisfazer todos os requisitos do comprador e, portanto, mais do que um fornecedor pode ser selecionado.

Durante o processo de seleção de fornecedores para a cadeia de abastecimento, que ocorre no início do fluxo como evidenciado na Figura 2.2, as empresas confrontam-se, geralmente, com um alto grau de incerteza e imprecisão. A teoria dos conjuntos difusos é considerado o método mais eficaz na gestão de problemas com incerteza. Para contornar os problemas de *multi-sourcing*, surgiu o desenvolvimento do **modelo integrado fuzzy TOPSIS e MCGP** (Liao e Kao, 2011, [3]).

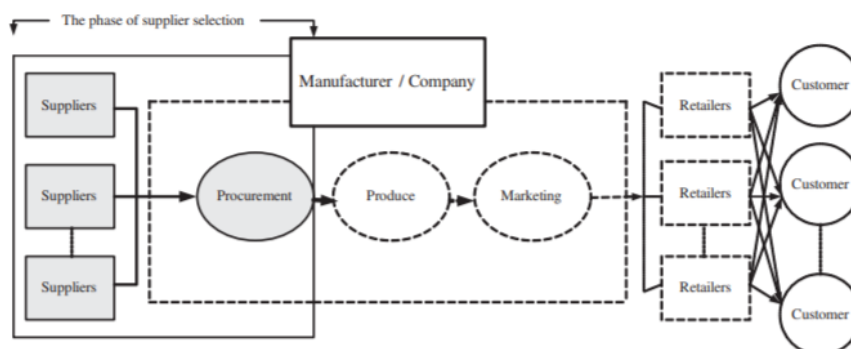


Figura 2.2: Processo de fluxo de gestão da cadeia de abastecimento em Chin-Nung Liao et al., «An integrated fuzzy TOPSIS and MCGP approach to supplier selection in supply chain management», [3], 2011

O TOPSIS é um método clássico do modelo MCDM e, como tal, pode fornecer a base para o desenvolvimento de modelos de seleção de fornecedores que pode lidar com incerteza. Tal abordagem baseia-se na ideia de que uma alternativa escolhida deverá ter a distância mais curta para a solução ideal positiva e a distância mais longa para a solução ideal negativa.

A importância atribuída aos vários critérios de seleção e as classificações obtidas de critérios qualitativos são consideradas variáveis linguísticas, sendo representadas através de funções lineares trapezoidais. O modelo não considera apenas a preferência dos decisores e a experiência para os critérios de seleção mas também inclui várias restrições tangíveis, como por exemplo, o orçamento do comprador, a capacidade dos fornecedores e o tempo de entrega.

A abordagem difusa da TOPSIS ajudou a converter a preferência e a experiência dos decisores (DM) em resultados significativos, aplicando valores linguísticos para avaliar cada critério e a relação com cada fornecedor.

A integração com o MCGP permitiu atribuir quantidades de pedidos a cada fornecedor, considerando o valor total criado a partir da aquisição. O MCGP possibilitou que os decisores definissem níveis de aspiração de múltipla escolha (MCAL) para cada objetivo com o intuito de evitar subestimação ou super estimação da tomada de decisão.

O algoritmo de tomada de decisão multi-pessoa e multi-critério através do método difuso TOPSIS e MCGP para lidar com a seleção de fornecedores é o seguinte:

1. Escolher as variáveis linguísticas apropriadas para a importância dos critérios de seleção e para as classificações linguísticas dos fornecedores;
2. Agregar o peso W_j do critério C_j e agrupar as classificações dos decisores para obter a classificação agregada difusa do fornecedor S_i sob o critério C_j ;
3. Construir a matriz de decisão difusa e normalizar a matriz;
4. Construir a matriz de decisão difusa normalizada ponderada;
5. Determinar o FPIS e o FNIS;
6. Calcular a distância de cada fornecedor ao FPIS e FNIS, respetivamente;
7. Calcular o coeficiente de proximidade de cada fornecedor;
8. De acordo com os coeficientes obtidos no passo 7 para cada fornecedor, construir o modelo integrado para encontrar os melhores fornecedores e as suas quantidades ótimas de pedidos. Para encontrar as melhores quantidades de pedidos, o valor total criado de aquisição (TVP) deve ser maximizado.

Estes modelos podem ser resolvidos usando a ferramenta LINGO onde serão mostradas as soluções ótimas.

2.4.4 Redes Neurais Artificiais

Trata-se de uma abordagem de modelação poderosa com aplicação em cadeias de abastecimento com redução significativa de custos e com melhorias no serviço prestado ao cliente. Na sua materialização incorpora todos os custos da cadeia, incluindo compra, produção, armazenagem, inventário e transporte.

Para além de se tratar de uma estratégia de planeamento, as organizações podem ganhar vantagem competitiva com a adoção desta metodologia uma vez que podem avaliar e implementar mudanças face a novas dinâmicas que vão sendo introduzidas numa organização, como introdução de novas rotas, novos parceiros, novos produtos, novas leis tributárias, entre outros.

2.4.4.1 Aplicações

Silva *et al.* (2017, [12]) aplicou o conceito de rede neuronal artificial de modo a **prever a capacidade** de uma cadeia de abastecimento em **atender pedidos** recebidos, assim como antecipar quais os nós da cadeia que receberão um pedido no próximo período. A ideia era compreender a forma como as redes neurais artificiais podem ser aplicadas para reduzir a vulnerabilidade de uma cadeia de abastecimento.

A abordagem seguida combinou um modelo de simulação de uma cadeia de abastecimento com uma arquitetura de rede neuronal artificial. O *software* de simulação utilizado para modelar a SC e definir os parâmetros a serem usados para alimentar a rede neuronal artificial foi o Simul8®. A rede neuronal artificial foi desenvolvida usando programação Matlab.

O modelo de simulação contemplou um único produto e consistia em quatro nós: fornecedor, produção, distribuição e retalhista. A rede neuronal foi desenvolvida com o intuito de prever o futuro a partir de dados históricos e, para isso, a sua configuração foi definida com três camadas: uma camada de entrada, uma camada de saída e uma camada oculta. O número de neurónios nas camadas de entrada e saída não eram fixos, sendo reajustados consoante os limites do passado e futuro. Os neurónios de entrada representavam os níveis de *stock* de cada nó da SC: retalhista, distribuição, produção e fornecedor.

Os dados gerados pela simulação ensinavam a rede neuronal para que a mesma adquirisse a capacidade de reconhecer e extrapolar para eventos futuros a partir de dados novos e não treinados. Os resultados foram obtidos para diferentes períodos de previsão e analisados perante um número crescente de períodos à entrada através da taxa de reconhecimento que traduz a eficácia da rede em prever eventos futuros.

Hui e Choi (2016, [21]) afirmou que no negócio de retalho de moda é essencial recolher uma grande quantidade de dados relacionados com as vendas e o histórico de compras do cliente. Esses conjuntos de dados podem ajudar o retalhista a identificar comportamentos de clientes e a descobrir padrões e tendências de compra para melhorar a qualidade de atendimento ao cliente e, naturalmente, a sua satisfação. A **exploração de dados** é uma abordagem que contém principalmente algoritmos de análise e descoberta de dados para produzir padrões e/ou regras significativamente importantes sobre uma grande quantidade de dados.

As redes neurais artificiais surgem como uma ferramenta aplicada neste âmbito com o intuito de encontrar padrões ocultos a partir de dados fornecidos. Na maioria dos casos de exploração de dados, a rede neuronal artificial adota o algoritmo de retropropagação, conduzindo a aprendizagem numa rede neuronal *feedforward*, exemplificada na Figura 2.3.

M e R (2015, [5]) usou a ferramenta anterior para classificar os clientes em diferentes grupos com comportamentos de compra específicos observados durante um longo período de tempo. As variáveis de entrada para o agrupamento de clientes incluíam a frequência de compra do produto, a preferência do produto e compra regular e aleatória. As variáveis de saída para classificação dos clientes eram classes de clientes: alto valor, médio valor, baixo valor. Uma rede neuronal artificial de retropropagação de três camadas foi construída para modelar a relação entre variáveis. Após o

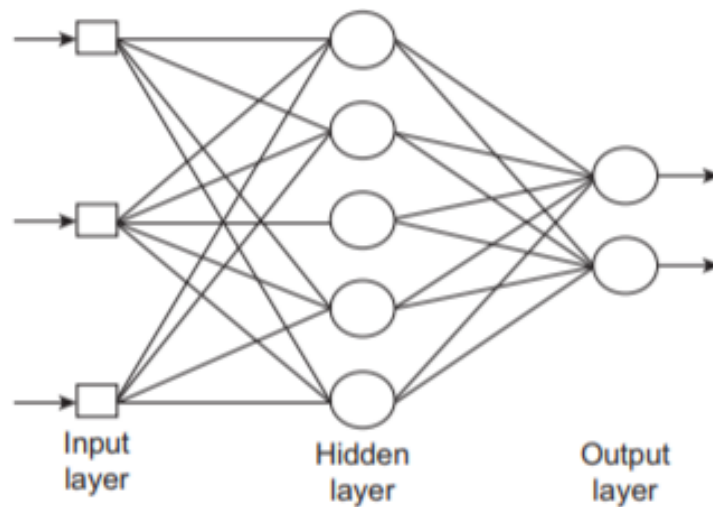


Figura 2.3: Rede neuronal *feedforward* multi-camada em P.C.L. Hui et al., [21], 2016

treino da rede com dados experimentais, verificou-se que a classificação do cliente era altamente precisa.

A **previsão de vendas** é uma tarefa muito difícil no negócio de moda uma vez que a procura de produtos é louca e imprevisível, o que obriga à existência de uma grande quantidade de *stock*. Outros fatores interferem, como a sazonalidade, estratégias de *marketing* e até mesmo o clima.

A previsão tradicional de vendas de moda baseia-se em métodos estatísticos comumente vistos, como a regressão linear, média móvel, média ponderada, suavização exponencial, suavização exponencial dupla. No entanto, tais métodos estatísticos podem não conseguir alcançar um resultado de previsão desejável (Hui e Choi, 2016, [21]).

Frank et al. (2003, [16]) realizou um estudo onde um modelo de rede neuronal artificial é utilizado para previsão de vendas de moda. A arquitetura de rede neuronal *feedforward* de três camadas foi implementada para aprender os padrões de vendas de peças de vestuário do passado para previsões de vendas no futuro.

Os resultados computacionais comprovaram que a rede neuronal artificial apresenta melhor desempenho em comparação com vários outros modelos de previsão estatística, sendo uma ferramenta eficaz também na previsão de vendas de moda.

2.5 Roteamento 4PL

Huang et al. (2015, [1]) afirmou que o **problema de otimização de roteamento na quarta camada logística (4PLROP)** é fundamental e inevitável na operação atual de um 4PL. Sob condições de emergência, a descrição de parâmetros incertos implica grandes desafios devido à falta

de dados históricos. Huang *et al.* (2015, [1]) estudou o problema de otimização de roteamento logístico com tempo de entrega incerto sob condições de emergência.

O 4PLROP sob condições de emergência foi formulado com base na teoria da incerteza, em que o tempo de entrega foi descrito como uma variável incerta. Posteriormente, o **modelo de programação incerto (UPM)** proposto foi comparado com o **modelo de programação estocástico (SPM)** no qual o tempo de entrega incerto foi descrito como uma variável estocástica baseada na teoria da probabilidade.

Este tipo de problema pode ser definido como a seleção de empresas logísticas para otimização da cadeia. Um problema chave que uma empresa 4PL deve considerar é a otimização de rotas de transporte de mercadorias desde a origem até ao destino final para minimização de custos.

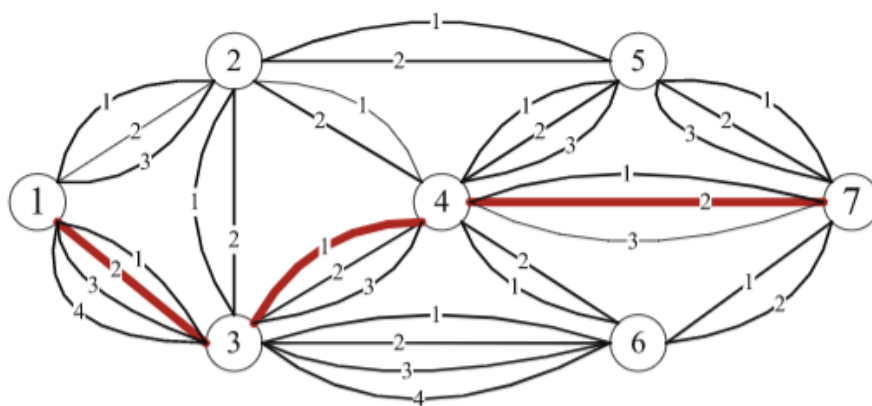


Figura 2.4: Multigrafo para problema de otimização de roteamento em Min Huang *et al.*, «4PL routing optimization under emergency conditions», [1], 2015

Como mostra a Figura 2.4, a rede de transporte pode ser apresentada sob a forma de um multigrafo constituído por nós e arcos. Os nós representam a origem, o destino mas também armazéns de material, locais de venda ou mesmo cidades, sendo estes denominados por pontos intermédios. Cada nó tem propriedades de custo e tempo.

Cada arco representa uma empresa 3PL que deseja realizar uma parcela da tarefa entre pares de nós. Tal como acontece com os nós, também os arcos têm propriedades de custo e tempo necessários para que a empresa 3PL complete o serviço de transporte entre dois nós. Entre pares de nós pode existir mais do que um 3PL para realizar a tarefa, sendo o tempo de entrega de cada um obtido de especialistas e considerado uma variável incerta.

2.5.1 Modelo de Programação Incerto (UPM)

O modelo matemático para o problema de otimização (4PLROP) considerou a variável tempo de entrega como sendo composta por duas parcelas: uma constante e outra variável. A parcela

variável seguiria uma distribuição de incerteza normal com um determinado desvio padrão e variância, cujo nível de confiança entre 0,5 e 1 assegurava a entrega de produtos de emergência.

O objetivo da solução para este modelo era encontrar uma rota origem-destino, incluindo nós e arcos, como evidenciado a vermelho na Figura 2.4, cujo custo fosse mínimo. Para que a função objetivo fosse alcançada, um conjunto de restrições relevantes foram tidas em consideração, por exemplo, o tempo total da rota não deveria exceder o tempo limite para entrega.

2.5.2 Modelo de Programação Estocástico (SPM)

De facto, a única diferença do SPM para o UPM residia na parcela variável do tempo de entrega que, no modelo de programação estocástico, era considerada uma variável aleatória que seguia igualmente uma distribuição normal incerta.

A forma como a teoria da incerteza e a teoria da probabilidade transformam um problema indeterminado num problema determinístico no âmbito do roteamento logístico é similar. No entanto, o resultado é diferente. Isto ocorre uma vez que a combinação linear para uma distribuição normal independente de variáveis incertas é diferente do caso em que estamos perante variáveis aleatórias.

Ambos os modelos determinísticos eram modelos de programação inteira e, como tal, foram resolvidos através do *software* CPLEX.

Huang *et al.* (2015, [1]) efetuou várias experiências em torno do 4PLROP para comparação dos modelos enunciados nas subsecções 2.5.1 e 2.5.2. Como referido na secção 2.5, o 4PLROP pode ser representado através de um multigrafo e, portanto, vários exemplos de multigrafos foram gerados aleatoriamente (número de nós, número de arcos entre dois nós, custos por nó e por arco, tempo por nó e por arco).

Os resultados detalhados para cada exemplo mostraram que os dois modelos podem ter diferentes soluções ótimas sob as mesmas condições. A solução do modelo UPM revelou-se mais robusta sob condições de emergência e mais adaptável para o 4PLROP proposto quando comparada com a solução do modelo SPM. O modelo UPM seleciona rotas com menor valor esperado de tempo e maior tempo de *buffer* do que o modelo SPM, o que implica custos mais elevados. No entanto, isso não significa que se trate de uma solução menos boa, mas sim evitar o risco de exceder o tempo limite na restrição.

Huang *et al.* (2015, [1]) constatou igualmente o efeito que o desvio-padrão e o nível de confiança produzem sobre a solução final, concluindo que o aumento de cada um individualmente provoca um aumento nos custos logísticos. A eficácia do método proposto ficou assim comprovada em problemas de roteamento logístico.

2.6 Conclusões

Além de ajudar na integração das atividades internas e na gestão da logística, uma cadeia de abastecimento identifica novas oportunidades de negócio, empreendedorismo, melhora os serviços e reduz custos operacionais.

Embora existam diversos métodos para otimização de cadeias de abastecimento, os problemas reais são, em regra, muito complexos de serem resolvidos, visto que grandes instâncias necessitam de um grande tempo de execução computacional e, muitas vezes, não se encontram soluções.

Dada a importância da redução de custos logísticos através da otimização e a complexidade de problemas com instâncias reais, torna-se essencial perceber o modelo de negócio em questão e consequentemente o tipo de cadeia em que se insere. São muitos os problemas e particularidades que um negócio e a cadeia envolvente vão enfrentando e muito por culpa da evolução do mercado. Como resposta e com o intuito de manterem vantagem competitiva, novas estratégias são pensadas e, na maioria dos casos, implementadas.

A pergunta que se impõe é a seguinte: qual a melhor estratégia e/ou método de otimização a aplicar? A resposta a esta pergunta depende de diversos fatores, entre eles, e talvez o principal, a natureza do assunto que se pretende otimizar na cadeia. Será a seleção de fornecedores? O nível de *stock*? A previsão de vendas? A produção em função da procura? Respondida a questão, a seleção pode recair sobre um simples modelo de programação linear ou não linear, metaheurísticas, lógica difusa, redes neuronais, modelos incertos ou estocásticos.

As redes neuronais e a lógica difusa revelam ser os métodos mais utilizados, embora todos os outros tenham aplicações e resultados alcançados. No entanto, estes últimos dois em comparação com os restantes mostram maior capacidade de análise de informação de uma forma mais rápida e precisa, partindo de um modelo altamente integrado.

Capítulo 3

Situação Atual da *Farfetch*

O modelo convencional de uma cadeia de abastecimento tem como principal foco apoiar o fluxo de materiais e informações ao longo de toda a cadeia mas também a conexão de todas as funções de negócio, embora conduzidas por diferentes entidades. A primeira etapa é, por natureza, a aquisição, seguindo-se a produção, a distribuição e as vendas ao cliente. A sequência destas operações está ilustrada na Figura 3.1.

A cadeia de um *e-commerce* é uma instância especial da cadeia de abastecimento convencional no sentido em que a sequência das operações é diferente. As vendas precedem a distribuição. No caso da *Farfetch*, as principais atividades ocorrem, precisamente, nestes dois processos.

Para além da *Farfetch*, também as lojas, as empresas transportadoras e os próprios clientes desempenham funções importantes nestes processos sendo, por isso, os grandes motores da cadeia, como evidenciado na Figura 3.2.

O crescimento rápido e sustentável desta retalhista de moda aliado ao aumento de pedidos exige um controlo rigoroso de todas as operações envolvidas na cadeia, bem como um acompanhamento permanente de todas as entidades comprometidas no serviço de fornecimento. Para perceber quem são essas entidades e a forma como se relacionam, é essencial detalhar o fluxo de negócio da *Farfetch*.

3.1 Posicionamento da *Farfetch*

O modelo de negócio da *Farfetch* permite que se diferencie de qualquer outro *website* de comércio eletrónico de moda. Para além de uma plataforma *e-commerce*, é também considerada um facilitador multi-canal para lojas de moda de todo o mundo. Com inúmeras coleções de lojas combinadas, a gama de marcas que oferece é mais ampla e variada do que em *websites* com um modelo de compra tradicional.

O foco da *Farfetch* é, não só, lojas com renome e influentes mas também lojas menos conhecidas e emergentes que destacam jovens *designers* com potencial. É esta mistura de diferentes classes de lojas que faz da plataforma um destino para a descoberta de novos talentos ao lado de marcas consagradas.

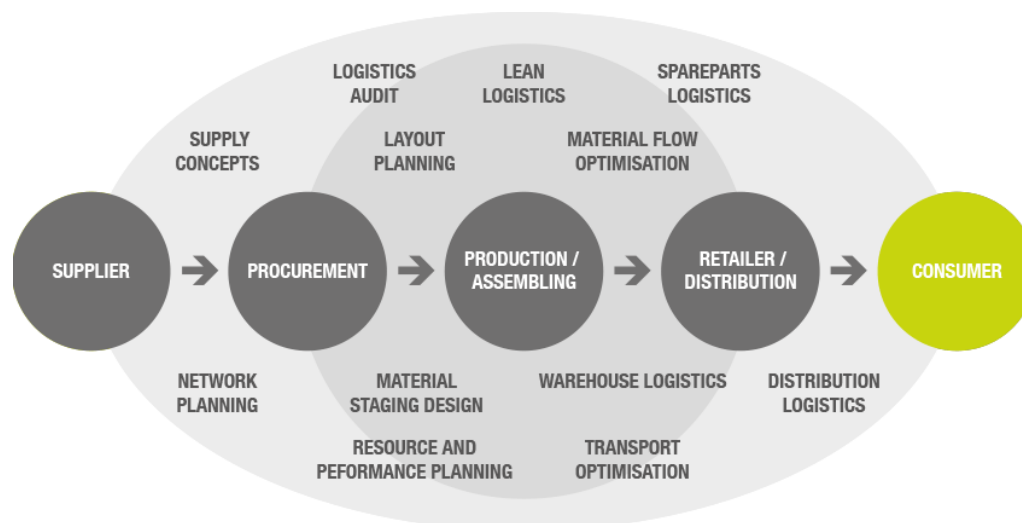


Figura 3.1: Fluxo tradicional da cadeia de abastecimento

3.2 Parceiros

A cadeia de abastecimento da *Farfetch* é caracterizada por uma comunidade global composta por cerca de 600 lojas e 200 marcas de luxo que oferecem uma experiência de compra inspiradora para os consumidores deste setor de luxo.

Cada pedido efetuado no portal da plataforma é atendido por um dos parceiros na sua loja física. As encomendas são tratadas com o máximo de detalhe e atenção, algo que apenas uma loja independente pode oferecer aos seus clientes. O objetivo é oferecer ao consumidor a experiência de ter visitado essa loja pessoalmente.

As lojas estão localizadas por todo o mundo mas unidas através de um único *website* - *Farfetch.com*. Com a *Farfetch.com*, qualquer cliente pode efetuar compras em *boutiques* de Paris, Londres, Nova Iorque, Milão, Los Angeles, entre outras, em apenas alguns minutos. Algumas das *boutiques* são de nomes muito respeitados - *L'Eclaireur*, em Paris, e *Feathers*, em Londres, por exemplo, que vendem algumas das marcas de moda mais conhecidas do mundo.

A rede da cadeia inclui também lojas especializadas em *designers* de moda emergentes, como é o caso da *B-store* em Londres e da *Henrik Vibskov* em Copenhaga. Existem ainda lojas conceituais cujo foco é a junção de marcas que construíram seguidores de culto. *Société Anonyme* em Florença e *Wok* em Milão são bons exemplos.

3.2.1 Operações dos Parceiros

A *Farfetch* não possui armazéns e, como tal, todos os *items* são geridos pelos parceiros. Os parceiros são responsáveis pelo processamento de pedidos, embalagem, gestão de preços e gestão de *stocks*, utilizando uma plataforma desenvolvida pela *Farfetch*, denominada de *STORM*, que permite aos parceiros registarem os seus pedidos e produtos.

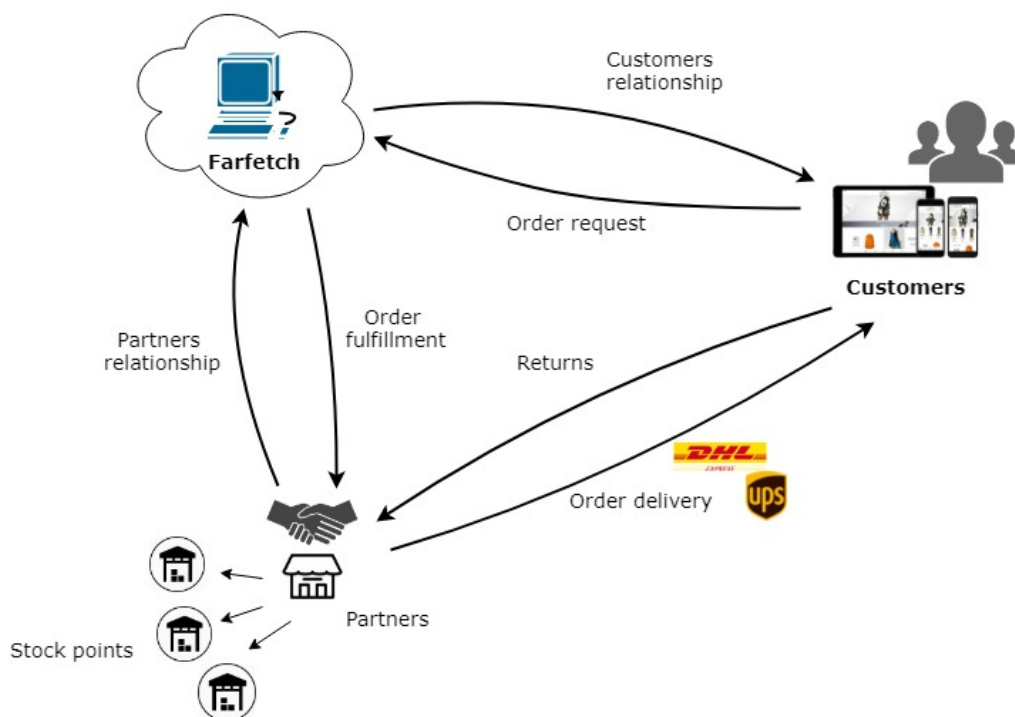


Figura 3.2: Representação do modelo de negócio da *Farfetch*

Existe um grande esforço por parte da *Farfetch* em permitir que todos os parceiros possam utilizar as ferramentas disponibilizadas por si, e um esforço ainda maior na integração das plataformas. Com esta integração, a sincronização entre lojas e plataformas *Farfetch* é mais fácil, fornecendo resultados mais precisos. Isso não invalida que, por vezes, não aconteça a situação de uma loja vender um artigo para um cliente físico e, simultaneamente, outro cliente o pretenda comprar através do portal.

Como forma de premiar os seus melhores parceiros e penalizar aqueles que se mostram menos comprometidos, a *Farfetch* projetou um programa de penalidades e incentivos, cuja versão atual utiliza métricas como o *No stock* e o *Speed of sending*, categorizando os parceiros em três categorias: negativa, neutra ou positiva. Na próxima versão a desenvolver, estas duas métricas serão correlacionadas e será tida em conta uma nova métrica, a do *item* errado. Outros importantes indicadores de desempenho são o *Time in transit* e o *Fraud payment refused*.

Para que uma loja possa exibir e vender um determinado artigo na plataforma *Farfetch.com* terá de enviar um exemplar desse produto para o centro de produção da *Farfetch*, em Guimarães. Neste processo de produção, os *items* são fotografados, as fotografias editadas e após aprovação, o artigo é devolvido à loja.

As lojas garantem a atualização de *stock* de cada produto que, posteriormente, fica disponível no *website*. Estas duas etapas são cruciais e dão o pontapé de saída para que o cliente efetue as suas encomendas e veja as mesmas serem satisfeitas, tal como ilustra a Figura 3.3.

É importante referir que os parceiros podem ter múltiplos *stock points* espalhados pelo mundo

para expedição de artigos, funcionando assim como armazéns, ou o caso em que o próprio parceiro é o *stock point*, como será explicado adiante.

3.2.2 Processamento de um Pedido

O tratamento de um pedido que entra no sistema da *Farfetch* envolve diversas fases sequenciais e algumas paralelas, existindo uma relação estreita entre parceiros e *Farfetch* também neste domínio, tal como exemplifica a Figura 3.3.

A partir do momento em que o cliente confirma um pedido no portal, todos os *items* são geridos pelos parceiros, estando do seu lado a responsabilidade pelo cumprimento do pedido. Os artigos pretendidos pelo cliente são associados a um *Portal order ID*, que designa a encomenda e, posteriormente, associados a diferentes lojas, dependendo da loja onde foram comprados e, como tal, são atribuídos identificadores específicos, designados por *Boutique orders*. É de salientar que o cliente não sabe em que loja está a comprar, apenas de que país vem o produto.

As lojas devem verificar a existência de um novo pedido que envolve os seus produtos e, para esse novo pedido, realizarem a validação de *stock*. Caso o *item* não esteja disponível, existe a possibilidade de reencaminhar o pedido para uma outra loja que possua *stock* ou de ser feita uma recomendação de um artigo semelhante, que o cliente pode ou não aceitar. Caso não aceite, o pedido é mesmo cancelado e o cliente recebe um *voucher*. Esta é considerada uma das piores experiências para o cliente, designada pela *Farfetch* como *No stock*.

Simultaneamente à validação de *stock*, ocorre a aprovação do pagamento realizada pela equipa de fraude da *Farfetch*. Nesta etapa, a equipa analisa se o cliente em questão é ou não fraudulento, verificando as suas contas sociais, endereços, número de telefone, entre outros detalhes. Se o cliente se revelar fraudulento, o pedido é de imediato recusado.

Caso o artigo pretendido se encontre disponível e o pagamento seja aprovado, o pedido avança para a próxima etapa, também ela da responsabilidade da loja: decidir a melhor embalagem de acordo com os *items* selecionados pelo cliente. As embalagens são entregues às lojas pela *Farfetch*.

Decidida a embalagem, a loja imprime a documentação necessária acerca da encomenda, nomeadamente a carta de porte, previamente criada pela *Farfetch*. O pedido é assinalado como pronto a enviar.

Algumas lojas têm *pick-ups* diários e, nesse caso, a encomenda será levantada na próxima recolha enquanto outras necessitam de combinar com a transportadora a data de levantamento do pedido. Os clientes recebem um *e-mail* assim que o pedido é enviado.

Se os clientes acharem que o(s) artigo(s) recebido(s) não se encontra(m) nas melhores condições, podem devolvê-lo(s). O processo de retorno inicia-se no momento em que o cliente solicita a devolução dentro do prazo estabelecido.

A transportadora inicia a recolha do pedido para devolução à loja, que deve confirmar as condições do pedido e decidir se o retorno deve ser aceite ou contestado. Caso o retorno seja aceite, a equipa de pagamentos da *Farfetch* processará o reembolso para o cliente.

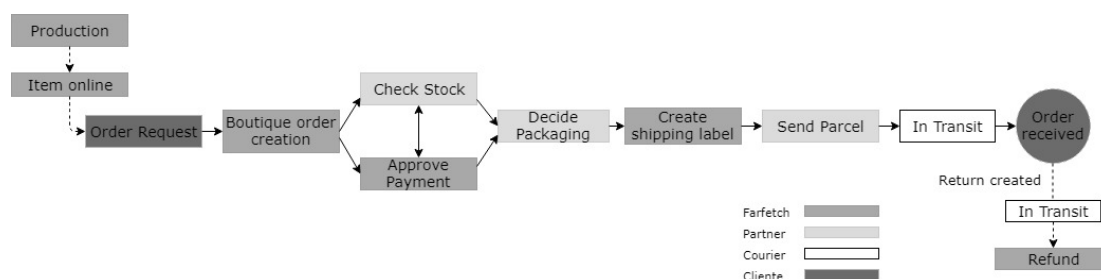


Figura 3.3: Fluxo principal de um pedido na cadeia de abastecimento da *Farfetch*

3.3 Stock Points

Os *stock points*, tal como o nome indica, são locais que os parceiros possuem com *stock* de produtos. Se os parceiros forem lojas, normalmente os *stock points* são as próprias lojas, o que significa que armazenam todos os artigos e estes são expedidos da própria loja. No entanto, se os parceiros forem marcas, os produtos são expedidos de locais de *stock* espalhados pelo mundo.

Estes *stock points* desempenham um papel muito importante uma vez que têm influência direta na decisão das rotas de envio. Existem várias regras para seleção dos melhores *stock points* que serão abordadas mais à frente no capítulo.

Uma das questões mais difíceis para esta retalhista de luxo é o seu esquema de transporte uma vez que lida com inúmeros *stock points* espalhados pelo mundo, ao contrário dos seus concorrentes comerciais que possuem armazéns e inventário, o que permite terem custos de envio mais baixos quando comparados com os da *Farfetch*.

3.3.1 Rotas

Devido à existência de *stock points*, as rotas de envio da *Farfetch* dividem-se em rotas diretas e rotas indiretas. As rotas diretas vão do *stock point* de origem diretamente para o endereço da carta de porte enquanto que as rotas indiretas podem ter múltiplas paragens até que a embalagem chegue ao destino final. Atualmente, este tipo de rotas são utilizadas no processo de devolução de produtos.

Uma rota pode ser definida pelo seu tipo de serviço, origem e destino. Informações adicionais incluem o fornecedor de serviços e a moeda em que a taxa fixa está definida. As rotas são estabelecidas com base na relevância dos países para o negócio, como ter uma loja parceira e um grande número de pedidos enviados ou por motivos operacionais. A *Farfetch* atualmente utiliza cerca de 300 rotas diferentes.

3.4 Transportadoras

Existem múltiplos serviços relacionados com a entrega das encomendas assim como transportadoras que realizam esses serviços. Se por um lado, a *Farfetch* fornece ao *checkout* opções e

custos de transporte tendo em conta regras de negócio, permitindo ao cliente a escolha do serviço que pretende, *Standard* ou *Express*, por outro lado, a escolha da transportadora para entrega da encomenda é também uma etapa fundamental.

O transporte tem duas categorias principais: *Metapack* e *Waybill*. A *Waybill* negocia diretamente com a UPS, DHL e com a única transportadora brasileira, os Correios. Já a *Metapack* é um serviço integrado, com mais de 400 transportadoras, cuja *Farfetch* negocia, principalmente, com os correios chineses: *China Post* e *SF Express*.

A *Farfetch* pode oferecer envio grátis dependendo do valor que o cliente gasta no portal, pagando esse custo à transportadora selecionada. Se um *item* for perdido ou danificado durante o transporte, a plataforma suporta o custo do produto para que a loja não seja afetada e o cliente receba o reembolso do artigo.

Para cada transporte é criada uma carta de porte que permite à transportadora atribuída levantar o *item* no ponto de *stock* indicado como origem nessa carta de porte.

3.5 Relação com o Cliente

A relação com o cliente desempenha um papel muito importante para a *Farfetch*. Não tendo inventário real nas suas mãos, a empresa tem de confiar nas operações dos seus parceiros. Apesar deste fator, a plataforma consegue estabelecer um bom relacionamento com o cliente através da recolha e análise de informação sobre o mesmo com o objetivo de criar uma estratégia de *marketing* individual. O envio de *e-mails* personalizados, por exemplo, faz parte dessa estratégia.

Os clientes têm à sua disposição o serviço de atendimento ao cliente através do qual podem expor dúvidas acerca de produtos, estado da encomenda, entrega ou devoluções. Este serviço pode ser personalizado pelo cliente em aspetos como o idioma, o tom de voz, o tipo de saudação, entre outros.

A *Farfetch* possui também múltiplas listas com informações acerca dos clientes, as chamadas *Whitelists*, *Blacklists* e VIP, sendo este um método de reconhecimento do cliente que está a efetuar uma compra no portal. As *Whitelists* são caracterizadas por clientes confiáveis para a *Farfetch* e que, na maioria dos casos, já compraram no portal sem problemas, revelando opções de pagamento seguras. No que se refere às *Blacklists*, os clientes são automaticamente rejeitados, quer por razões de histórico ou uso de cartões de crédito falsos. Os clientes VIP são aqueles que dispendem muito no portal e os seus pedidos são automaticamente aprovados.

3.6 Gestão da Cadeia

Como referido na secção 3.5, a *Farfetch* não possui qualquer inventário dos artigos que estão à venda no *website*, sendo este o maior desafio do seu modelo de negócio. Aquilo que acontece é um contacto permanente por parte de uma equipa de operações com os parceiros distribuídos pelo mundo para, desse modo, ser assegurada uma boa comunicação, essencial em qualquer cadeia de abastecimento.

3.6.1 Visão de Produto

O fluxo de um pedido inicia-se no portal *Farfetch.com* onde o cliente está em contacto com tudo aquilo que a plataforma oferece. Após o cliente selecionar os produtos que pretende adquirir, adicionando-os ao carrinho, segue-se o *checkout*.

Numa fase inicial, o pedido seguia diretamente para o serviço de gestão de encomendas (OMS), no entanto, com a implementação do serviço de roteamento, o pedido passa do *checkout* para o serviço de roteamento e, somente depois, para o OMS.

O serviço de roteamento, durante o seu processamento e ainda antes do pedido prosseguir para o OMS, comunica com o Atlas, um serviço centralizado responsável por filtrar a informação recebida e retornar todas as rotas que se ajustam a essas informações.

3.6.1.1 Checkout

O *checkout* é a secção que marca o início do processamento de um pedido pela *Farfetch* e o seu procedimento está atualmente ordenado e dividido nos seguintes passos:

1. *Shipping*

Na página de *shipping*, o cliente pode escolher o destino do seu pedido. O local da entrega pode ser um endereço que o cliente introduz ou um local de *Click & Collect*. Caso a escolha recaia sobre um local *Click & Collect*, isso significa que o cliente prefere levantar a sua encomenda num parceiro da *Farfetch* localizado perto de si.

No caso de inserir um endereço de entrega, o preço de transporte ajusta-se a esse endereço, ou seja, endereços de entrega diferentes, podem implicar preços de transporte diferentes.

2. *Payment*

A etapa de pagamento inclui a escolha do método de pagamento que também varia consoante o endereço introduzido pelo cliente no passo anterior. No entanto, as opções variam entre o método de *PayPal*, cartão de débito ou cartão de crédito.

Caso o cliente opte pelo método de pagamento com cartão de débito ou crédito, terá de introduzir informações sobre o cartão, ao passo que se selecionar o método *PayPal*, a *Farfetch* apenas necessita de saber qual o *e-mail* da conta *PayPal* do cliente.

Diferentes países podem ter diferentes métodos de pagamento.

3. *Review*

Na etapa de revisão, o cliente seleciona o método de entrega que pretende. Os serviços de entrega disponíveis são *Express* ou *Standard*. Além destes, podem também estar disponíveis os métodos *Same Day* e *F90*. Nesta secção e consoante o método escolhido, o cliente tem a indicação de uma estimativa da data de entrega do seu pedido, o chamado *Expected Delivery Date (EDD)*, uma informação pouco confiável e que pode falhar, dependendo de muitos fatores. O cliente consegue ainda visualizar o custo de transporte acrescido ao custo total do produto adquirido.

O custo de transporte está relacionado com o serviço de entrega selecionado e diferentes serviços podem ter diferentes transportadoras associadas.

3.6.1.2 Serviço de Roteamento

O serviço de roteamento comunica com o *checkout* para fornecer informações sobre os serviços de transporte disponíveis para um pedido específico, sendo responsável por selecionar o melhor *stock point* capaz de atender esse pedido. A decisão é tomada com base num conjunto de regras que estão implementadas e que vão de encontro às necessidades e preferências das marcas. As regras estão mapeadas através de um polinómio com ponderações associadas a variáveis específicas, equação 3.1, cujas métricas aplicadas são explicadas de seguida:

1. **HasStock:** Para cada *stock point* é verificado se tem *stock* suficiente para satisfazer o artigo. Caso a quantidade disponível seja superior à quantidade que o cliente deseja, retorna 1, caso contrário retorna 0. Trata-se, portanto, de uma variável binária de verificação de *stock*.
2. **Click & Collect:** O cliente tem a possibilidade de selecionar durante o *checkout* na etapa de *shipping* se pretende ir levantar a encomenda a um parceiro perto de si ou se insere o seu endereço para que lhe seja entregue em casa. *Click & Collect* não está disponível para todos os parceiros da *Farfetch*. Caso o cliente selecione a opção de *Click & Collect*, o serviço de roteamento recebe essa informação e leva-a em consideração na escolha do melhor *stock point* com uma ponderação de 10000.
3. **BrandRanking:** Dispara um ranqueamento consoante a preferência das marcas por país. A ponderação para esta variável é de 1000.
4. **BasketCoverage:** Verifica se um *stock point* pode entregar artigos diferentes de uma *portal order*. Se sim, esses artigos são alocados ao mesmo *stock point*. A ponderação fixada para esta variável é de 1000.
5. **MerchantFromRequest:** Esta variável atribui preferência ao *stock point* alocado através do algoritmo de duplicados em *front-end*. Não sendo a variável com mais impacto para a decisão final, é-lhe atribuída uma ponderação de 10.

No caso de existir um empate entre parceiros, recorre-se à performance das lojas, traduzida através do NPS. O NPS é um índice que, no caso da *Farfetch*, varia entre -100 e 100, representativo da experiência do cliente no que se refere à recomendação dos produtos e/ou serviços prestados. É uma das formas de medir a satisfação do cliente perante o produto adquirido assim como a sua fidelidade à marca.

6. **MerchantRanking:** Dispara um ranqueamento de *stock points* que pode ser diferente de país para país.
7. **Stock Depth:** A profundidade de *stock* é também um fator a ter em conta na decisão final. Para um determinado artigo, é averiguada a quantidade disponível desse mesmo artigo nos

vários *stock points*. É dada prioridade àqueles que dispõem, naquele momento, de uma maior quantidade.

$$HasStock \times (Click\&Collect \times 10000 + BrandRanking \times 1000 + BasketCoverage \times 1000 + MerchantFromRequest \times 10 + MerchantRanking \times \frac{StockDepth}{10}) + \frac{MerchantFromRequest}{MerchantRanking \times 100} \quad (3.1)$$

3.7 Limitações do Serviço de Roteamento

O serviço de roteamento é uma fase crucial no processo de tratamento de pedidos. Tratando-se de uma etapa que não foi implementada desde início, aparece ainda com algumas limitações, o que significa que possui espaço para melhorias no seu funcionamento.

Atualmente, e como referido na subsecção 3.6.1.2, o funcionamento deste serviço tem como base um polinómio com uma série de variáveis cujo propósito maior é a atribuição de um *stock point* por artigo de modo a que o pedido seja satisfeito no seu todo. No entanto, é importante salientar que a empresa tem conhecimento que se trata de um algoritmo com uma complexidade baixa, o que se traduz na seleção de boas estimativas de *stock points* e não, em *stock points* ideais, como seria pretendido.

As restrições existentes no polinómio são de diversa natureza. Declara-se como um polinómio redutor, pouco abrangente, limitado a regras muito específicas que não permitem uma visão global do negócio, privilegiando apenas algumas métricas, o que desde logo impossibilita a *Farfetch* de avaliar e analisar a influência direta que outras métricas igualmente importantes teriam na escolha do *stock point*. Neste sentido, existe muita incerteza acerca dos impactos que as métricas atualmente inseridas no polinómio acarretam. O ideal seria poder antecipar esses impactos, algo que não é possível com o algoritmo atualmente implementado.

As ponderações associadas a cada uma das métricas avaliadas no polinómio foram definidas em bases de 10 consoante a importância a atribuir a cada uma na seleção de *stock points*. Trata-se de uma importância relativa definida pela empresa e, como tal, existe o conhecimento de que não são as ponderações ideais, apenas uma forma simplificada de valorizar variáveis em detrimento de outras no momento da decisão.

Para além do referido acima, é um polinómio que encerra pouca lógica de negócio na otimização, primando pelo seu lado mais operacional na busca de ações diretas e menos pelo seu lado estratégico na busca de direções a seguir.

Capítulo 4

Solução Proposta

Desde início do projeto ficou comprovado e esclarecido que o método atualmente utilizado pela empresa na escolha do melhor parceiro para satisfação de uma determinada encomenda, integrado no serviço de roteamento, possui algumas deficiências e limitações, tal como explicado no capítulo 3, mais especificamente na secção 3.7.

Nesse sentido, e com o objetivo de obter o maior lucro possível por pedido, garantindo que os parceiros continuam a vender, fornecendo ao cliente um serviço exclusivo e de luxo, tornou-se prioritário projetar e desenvolver uma estrutura que fosse capaz de acompanhar todo o processo, desde a definição e modelação de uma hipótese ou função objetivo, passando por um ambiente de simulação e avaliação, terminando no seu momento chave, a realização de uma regressão.

Idealmente, a estrutura deverá ser flexível e integrável com as plataformas da *Farfetch*, assegurando a diminuição do tempo de chegada ao mercado, ou se preferirmos, de colocação em *live*, bem como a garantia de uma melhoria contínua do algoritmo. Indo de encontro à revisão de literatura realizada no capítulo 2, as redes neuronais artificiais revelaram-se o caminho certo a seguir neste âmbito.

4.1 Visão Geral da Estrutura de Otimização

A estrutura delineada foi pensada e arquitetada de modo cíclico, sendo caracterizada por um conjunto de etapas sequenciais e dependentes entre si que permitam alcançar os objetivos inicialmente propostos, como exemplifica a Figura 4.1.

Numa primeira fase, revelou-se essencial o estudo acerca das métricas a considerar na modelação do problema, sendo esta uma etapa importante que desencadeia e permite o funcionamento da restante estrutura. É relevante realçar que a execução desta etapa foi realizada em parceria com a equipa de operações da *Farfetch*, nomeadamente com um outro estudante da Faculdade de Engenharia que se encontrava, também ele, a realizar a sua dissertação de mestrado no âmbito do curso de Engenharia e Gestão Industrial.

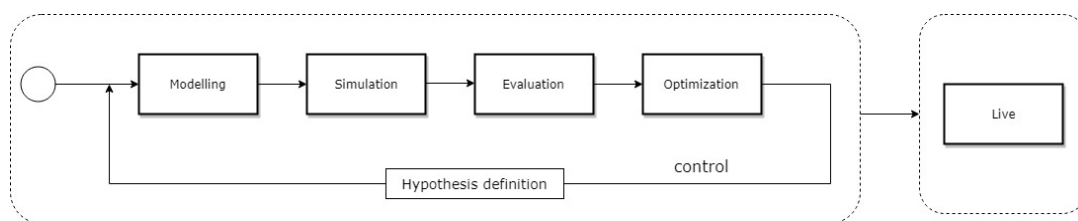


Figura 4.1: Visão geral da estrutura de otimização desenvolvida

A segunda etapa, a simulação, tem como ponto de partida a etapa anterior, a modelação, para estimação das métricas definidas com base no histórico de encomendas da *Farfetch* para um determinado intervalo temporal. Segue-se a avaliação, uma etapa que pretende analisar os resultados obtidos numa perspetiva de comparação com valores que seriam os ideais, estes definidos e fornecidos pela equipa de operações.

Medinte a análise efetuada e os resultados obtidos, a estrutura entra na sua etapa mais delicada e fulcral, a regressão. Nesta fase, a hipótese inicialmente modelada é reestruturada com o intuito de favorecer ou penalizar as métricas que a definem, assegurando a melhoria contínua do algoritmo. Encontrada a solução ótima para a hipótese definida, estão reunidas as condições para a seleção do melhor *stock point* capaz de satisfazer determinado pedido do cliente.

O ciclo da estrutura de fecha-se com o controlo da definição da hipótese, um controlo que deverá ser efetuado também em conjunto com a equipa de operações, existindo um contacto permanente no sentido de acompanhar o estudo de novas métricas para futura inserção e exploração na estrutura desenvolvida. A partir daqui, o processo volta à sua etapa inicial, reiniciando-se o ciclo.

A disposição em *live* da estrutura implementada é, desde início, o principal objetivo da empresa, um passo que depende de vários fatores, daí a dimensão e valia que a estrutura deve adquirir. Para além disso, trata-se de um período que pode demorar algum tempo uma vez que não é linear.

4.2 Modelação

Dentro de um ambiente dinâmico e complexo como o previamente identificado e tratando-se de um problema de otimização multivariável e multiobjetivo que deve estar alinhado com a visão estratégica da empresa, a modelação do problema foi, desde logo, o momento mais demorado do projeto uma vez que exigia um conhecimento alargado e consciente acerca do modelo de negócio da *Farfetch*, quer em termos operacionais, quer em termos estratégicos. Nesse sentido, foram identificadas três grandes perspetivas para atuação em que assenta o seu modelo de negócio:

1. **Lucro:** São milhares as encomendas diárias que a *Farfetch* recebe. Mas afinal, que parte consegue a *Farfetch* ganhar com a venda de produtos através do seu *marketplace*? Que custos diretos e indiretos estão associados a esta venda? Que impacto têm no valor final de venda do produto? Existe lucro em todas as encomendas? Ou algumas causam prejuízo?

2. **Cientes:** Qualquer pessoa, em qualquer parte do mundo, pode comprar na *Farfetch*, bastando para isso que aceda ao portal *FF.com*. Sem clientes, não há crescimento e, como tal, assegurar a satisfação do cliente é algo indiscutível e desafiante.
3. **Parceiros:** Os grandes aliados da plataforma *e-commerce* que se comprometem a cumprir os pedidos dos clientes. Do lado da *Farfetch* interessa perceber como vão as suas vendas, garantindo o seu crescimento através da alocação dos pedidos dos clientes.

4.2.1 Lucro

Quando mencionamos o conceito de lucro na *Farfetch*, e tal como acontece em qualquer outra empresa dentro ou fora do setor, referimo-nos à totalização de todas as fontes de receita e subtração de todas as despesas, inferindo a evolução financeira da organização num determinado intervalo de tempo. Tão ou mais importante do que esse cálculo, está uma análise crítica e aprofundada de todas as métricas de *e-commerce* consideradas chave para uma visão geral do negócio e estudo da lucratividade num determinado período.

Desse modo, para a perspetiva de lucro foram consideradas variáveis diretamente ligadas a esse propósito ou que, de alguma forma, interferem no rendimento obtido a partir das vendas: comissões, *geomargin* e custos de transporte.

Comissões

As receitas angariadas pela empresa são alcançadas, sobretudo, graças às comissões aplicadas sob todas as vendas bem sucedidas. As políticas de comissão dependem das condições de contrato com cada marca, o que quer dizer que o seu valor pode variar. A comissão depende da origem do ponto de *stock* (loja). Por exemplo, para uma determinada marca, todas as vendas provenientes de lojas dos EUA têm uma percentagem de comissão aplicada diferente. No entanto, aquilo que se verifica é que o seu valor médio de entre todos os parceiros está entre os 20 e os 25%.

Se considerarmos o preço definido pela loja para cada artigo, designado por *Recommended Retail Price*, cuja origem está na soma do preço grossista com o aumento de preço proposto pela marca e lhe aplicarmos a percentagem de comissão respetiva, estamos perante o valor ganho pela *Farfetch* com a venda de determinado artigo, como mostra a equação 4.1.

$$\text{Valor de comissão} = \text{Percentagem de comissão} \times \text{Preço recomendado pela loja} \quad (4.1)$$

Geomargin

O preço a que um produto é vendido na loja não é, em grande parte dos casos, o preço a que é vendido no portal *FF.com*. Perante o preço definido pela loja, a *Farfetch* calcula através de algoritmos com múltiplos indicadores o preço final de venda, o chamado preço fixo do *item*. Para este cálculo são levados em consideração múltiplos aspetos dependendo do país a partir do qual o produto irá ser vendido, nomeadamente as suas políticas económicas, a moeda local e suas flutuações, entre outros.

Vejam os seguintes exemplos para que possamos entender melhor este mecanismo: imagine que um artigo irá ser vendido a partir de uma loja no Reino Unido para um cliente nos Estados Unidos. O preço recomendado pela loja é de £1025. A este valor será retirado o imposto sobre o valor acrescentado (VAT) de bens ou serviços locais, o que significa que, estando a loja situada no UK, esse valor está fixado nos 20%. Pelo contrário, é adicionado o chamado subsídio de envio, criado quando a taxa fixa de envio é menor que a usual, o que leva a uma redução dos custos de envio mas a um aumento do preço. Na maioria dos países, esse aumento é de 3,76%.

Nas situações em que a moeda de compra do cliente é diferente da moeda local da loja, torna-se necessário realizar a conversão, neste caso de libras (GBP) para dólares (USD), aplicando a taxa de conversão definida pela *Farfetch* para a transação em questão. É de notar que as taxas de conversão são atualizadas pela empresa mensalmente. Devido às flutuações das moedas locais, a *Farfetch* criou uma margem que é mais ou menos elevada consoante a maior ou menor flutuação. No caso da Rússia, por exemplo, existe uma margem de 10% uma vez que a sua moeda, a *Russian Ruble*, flutua bastante em comparação com outras, por exemplo, o Brasil, onde devido às altas taxas alfandegárias, essa margem não existe.

O último passo até ser atingido o preço final é a inclusão dos impostos. Dependem, fundamentalmente, do país de destino onde a encomenda será entregue, sendo determinados pela agência alfandegária desse mesmo país.

Estes impostos podem ser apresentados de duas formas diferentes: DDP (taxas de entrega pagas) ou DDU (taxas de entrega não pagas). Se o país for DDP, o preço final incluirá o valor do *item*, transporte e impostos, enquanto que, se o país for DDU, os impostos que o cliente tem de pagar não estão incluídos no preço final. Neste último caso, o cliente terá de pagar quando receber a encomenda no seu país, recebendo uma fatura com esse valor.

A diferença de preços, ilustrada na equação 4.2, pode ser negativa, no caso em que o preço final de venda é inferior ao preço definido pela loja e, quando assim é, o parceiro cobre essa diferença, ou positiva e, dependendo da versão do contrato, poderá receber a *Farfetch*, o parceiro ou ambos. Esta diferença é designada pela empresa de *geomargin*.

$$\text{Geomargin} = \text{Preço final de venda} - \text{Preço recomendado pela loja} \quad (4.2)$$

Custos de Transporte

O transporte assume um papel importante para a *Farfetch*, fazendo parte do processo de cumprimento de um pedido. O custo associado ao envio de uma encomenda vem, sobretudo, de acordos realizados com as transportadoras. Para cada uma delas são definidos preços distintos conforme o tipo de serviço que realizam e a rota origem-destino.

Caso a transportadora selecionada para entrega de encomendas seja a DHL, a *Farfetch* recorre a uma ferramenta da própria transportadora disponível no seu *website* que permite o cálculo do chamado peso volumétrico ou dimensional, com base nas dimensões da caixa que foram recomendadas à loja responsável pela satisfação do pedido. A ferramenta está representada na Figura 4.2. Trata-se de um peso estimado que, nem sempre, corresponde ao peso real. Com base nesse peso

estimado e na rota origem-destino atribuída ao pedido, a empresa determina o custo de transporte que terá de pagar à transportadora. Desta forma, os custos de transporte relativos à transportadora DHL com serviço *Express* foram calculados a partir da equação 4.3:

$$\text{Custos de transporte} = 1.1175 \times \text{values} \quad (4.3)$$

onde 1.1175 representa a taxa média de câmbio e *values* as conversões realizadas da moeda do cliente para a moeda da loja.

Tipo de Embalagem ☒ Caixa ☐ Tubo

Unidade de medida ☒ Cm / Kg ☐ Polegadas / Libras

Largura cm

Comprimento cm

Altura cm

Peso Volumétrico kg

Figura 4.2: Cálculo do peso volumétrico

O peso real, esse, apenas é averiguado pela transportadora no momento em que a caixa chega às suas mãos. Existem situações em que o peso real está acima do peso volumétrico, o que leva a *Farfetch* a cobrir essa diferença para além do estimado inicialmente. Caso se verifique o contrário, em que o peso real é inferior ao peso volumétrico, a DHL devolve à empresa o montante respetivo.

Se a transportadora em causa for a UPS, a situação é distinta, até porque existem diversas variantes desta transportadora: *UPS Standard* e *UPS Ground* que realizam serviço *Standard* e *UPS Second Day Air* e *UPS Next Day Air Save* que realizam serviço *Express*.

No caso da *UPS Ground* e da *UPS Next Day Air*, apenas realizam serviço dentro dos Estados Unidos, o que implica que sejam realizados tendo em conta os chamados *zipcodes* de origem, respeitantes aos *stock points*, e de destino, associados aos clientes, em detrimento do peso da caixa que deixa de ter influência no cálculo do custo.

Os *zipcodes* não são mais do que códigos postais, quer do cliente, quer do *stock point*, em que os três primeiros caracteres de cada *zipcode* determinam a zona a que pertence a combinação.

Vejamos um exemplo: o *stock point* para expedição da encomenda fica algures na Austrália cujo código postal é dado por 2092, enquanto que o cliente está situado algures na Alemanha com um código postal de 88131. Ora, a combinação (209, 881) tem como zona atribuída, a zona 203. A partir da zona a que pertence a combinação de *zipcodes*, calcula-se o custo de transporte a partir

da equação 4.4:

$$\text{Custos de transporte} = 0.8 \times \text{values} \quad (4.4)$$

onde 0.8 representa a taxa média de câmbio considerada pela equipa de operações e *values* as conversões de moeda.

No caso da UPS *Standard*, as rotas abrangem apenas a Europa enquanto que a UPS *Next Day Air Save* atua sobretudo pela Europa e USA. Como tal, o cálculo do custo de transporte para estes casos foi efetuado da mesma forma que para os casos da DHL em que o peso da caixa assume um papel preponderante.

A perspetiva de lucro foi assim caracterizada e modelada pelas três variáveis abordadas anteriormente e que se traduzem na equação 4.5:

$$\text{Lucro (\$)} = \text{Comissão} + \text{Geomargin} - \text{Custos de transporte} \quad (4.5)$$

4.2.2 Parceiros

Tendo o modelo de negócio da *Farfetch* uma grande expressão junto dos seus parceiros, parece evidente a necessidade absoluta em aferir o seu desempenho, sobretudo em termos de vendas, de forma a assegurar o seu crescimento sustentável. As lojas possuem dois tipos de vendas: as vendas efetuadas no portal da *Farfetch* e as vendas *offline*, aquelas que são realizadas a partir da loja física. Estes parceiros podem ser avaliados através de duas métricas um pouco distintas: o *Sell Through* e o peso das vendas *online*. Naturalmente, as vendas que são realizadas através do portal *FF.com* ganham uma dimensão superior para a *Farfetch* em comparação com as vendas *offline* uma vez que interessa à empresa que um parceiro consiga vender em números superiores através do *website*, retirando assim maior benefício.

Sell Through

O *Sell Through* é um indicador medido em percentagem e em períodos semanais de domingo a sábado, sendo representado pela razão entre o somatório das vendas *online* com as vendas *offline* e o total de *stock* fornecido ao parceiro no início da semana, tal como ilustra a equação 4.6:

$$\text{Sell Through} = \frac{\text{vendas online} + \text{vendas offline}}{\text{stock recebido}} \quad (4.6)$$

Trata-se de uma métrica que pode ser analisada a partir de diversos aspetos: por produto, por loja, por estação ou até mesmo por categoria ou família de produto. As principais estações consideradas são: Outono-Inverno e Primavera-Verão. Para além destas designações existem ainda artigos *Vintage* que não passam de moda, que perduram no tempo, artigos designados por *Carryover* que passam de uma estação para a outra e artigos *Archive* que já não estão no portal da *Farfetch* por não venderem *online*.

As categorias de um produto são inúmeras e diversificadas, atualmente cerca de 200 e estão incluídas em famílias. Uma família pode abranger várias categorias. Calções, jaquetas, botas, cintos são apenas quatro dessas cerca de 200 categorias. Os calções e jaquetas pertencem à família de roupa, as botas à de calçado e os cintos aos acessórios.

Peso das Vendas Online

O peso das vendas *online* é igualmente uma métrica medida em percentagem e em períodos semanais e que, tal como o próprio nome indica, permite avaliar em que proporção as vendas realizadas no portal da *Farfetch* se superiorizaram em relação às vendas realizadas na loja. O seu cálculo é efetuado de acordo com a equação 4.7:

$$\text{Peso das vendas online} = \frac{\text{vendas online}}{\text{vendas online} + \text{vendas offline}} \quad (4.7)$$

A perspetiva de parceiros foi caracterizada e modelada pelas duas métricas abordadas anteriormente e a forma como foram relacionadas está expressa na equação 4.8:

$$\text{Parceiros (\%)} = \text{Peso das vendas online} \times \text{Sell Through} \times 100 \quad (4.8)$$

4.2.3 Clientes

Com o aumento crescente do número de pedidos, tornou-se crucial não só medir o desempenho das lojas, mas também o desempenho das transportadoras. Mesmo que a loja consiga cumprir com as exigências da *Farfetch*, se a transportadora não respeitar os horários acordados, a experiência do cliente sairá prejudicada e a *Farfetch* será responsável por isso. Nesse sentido, a empresa necessita de manter um controlo rigoroso sobre os tempos de entrega das transportadoras, a fim de garantir a satisfação total do cliente. Perante esta realidade, na perspetiva de clientes revelou-se fulcral incluir e avaliar a métrica do tempo de entrega.

Tempo de Entrega

Como sabemos, o conceito de tempo de entrega entende-se como o tempo total, em dias, que decorre desde que o cliente efetua um pedido até que o recebe. No modelo de negócio da *Farfetch*, um único pedido pode incluir diversos artigos, os quais podem ser alocados a diferentes *stock points* para posterior expedição. Caso se verifique tal cenário, aquilo que sucede é que todos os artigos terão tempos de entrega diferentes, o que quer dizer que o cliente receberá o seu pedido repartido, algo pouco desejável quando comparado com a hipótese de puder receber o seu pedido por inteiro e num único dia.

A empresa considera dois tipos de tempos de entrega: o tempo de entrega bruto e o tempo de entrega líquido. A única diferença existente entre ambos é o facto do tempo de entrega líquido considerar feriados e fins-de-semana e o tempo de entrega bruto apenas dias úteis. Na perspetiva de clientes, o foco incidiu sobre o tempo de entrega bruto, cujo cálculo precisa da compreensão de duas outras variáveis diretamente relacionadas: velocidade de envio e tempo em trânsito no passo

6.

Velocidade de Envio

A velocidade de envio é um indicador que pode ser calculado de duas formas distintas: considerar a data de criação da encomenda e a data em que a encomenda é levantada na loja, aplicando a equação 4.9. Ou então, avaliar os cinco primeiros passos do cumprimento de um pedido, desde a verificação de *stock* pela loja até ao levantamento por parte da transportadora, medindo o tempo dispendido em cada um dos passos referidos, permitindo dessa forma avaliar o desempenho de cada entidade envolvida, desde as lojas, à equipa de fraude da *Farfetch* e até o próprio sistema *Farfetch*. A escolha recaiu sobre a primeira forma de cálculo.

$$\text{Velocidade de envio} = \text{Data de levantamento na loja} - \text{Data de criação da encomenda} \quad (4.9)$$

Tempo em Trânsito no Passo 6

Trata-se de uma métrica essencial que afeta diretamente a satisfação do cliente e permite avaliar a eficiência das transportadoras. Este indicador é medido a partir do momento em que é feito o *pick-up* da encomenda na loja pela transportadora, até ao momento em que o cliente recebe o pedido, traduzindo-se pela equação 4.10. O tempo em trânsito no passo 6, assim designado este indicador, depende de diversos fatores, desde o tipo de transportadora, o serviço que realiza e a rota origem-destino do pedido.

$$\text{Tempo em trânsito no passo 6} = \text{Data recebida pelo cliente} - \text{Data de levantamento na loja} \quad (4.10)$$

O tempo de entrega bruto foi, assim, considerado como métrica chave para a perspetiva de clientes, sendo obtido pela soma da velocidade de envio com o tempo em trânsito no passo 6, como exemplifica a equação 4.11:

$$\text{Tempo de entrega bruto (dias)} = \text{Velocidade de envio} + \text{Tempo em trânsito no passo 6} \quad (4.11)$$

4.3 Definição da Hipótese

Posteriormente ao estudo e seleção de métricas a considerar para cada uma das perspetivas em avaliação, apresentadas na secção 4.2, segue-se a próxima etapa no desenvolvimento da estrutura de otimização, a definição daquilo a que designamos por hipótese, que não é mais do que um cenário em análise, traduzida através de um simples polinómio. A ideia principal é validar e avaliar o impacto que a hipótese definida causa nas três perspetivas consideradas chave para este projeto. Para que este objetivo seja atingido, o polinómio que compõe a hipótese deve englobar as três perspetivas abordadas: lucro, clientes e parceiros, como ilustrado na equação 4.12. Uma vez

que cada um dos pontos de vista tem a si aliado as respetivas métricas, a equação 4.12 transformou-se na equação 4.13, sendo este o polinómio final que serviu de base para a execução do restante sistema.

$$\text{Hipótese} = \text{Lucro} + \text{Parceiros} + \text{Clientes} \quad (4.12)$$

$$\begin{aligned} \text{Hipótese} = & (\text{Comissão} + \text{Geomargin} - \text{Custos de transporte}) + \\ & (\text{Peso de vendas online} \times \text{Sell Through}) + (\text{Tempo de entrega bruto}) \end{aligned} \quad (4.13)$$

As equações 4.12 e 4.13 espelham a afirmação referida na secção 1.2 de que, de facto, estamos perante um problema de otimização multivariável, com variáveis de diversa natureza, e multiobjetivo, já que pretendemos equilibrar três diferentes perspetivas como veremos mais à frente.

4.4 Recolha de Dados

Concluída a primeira grande fase do desenvolvimento da estrutura de otimização, a modelação, quer das métricas, quer do cenário para análise, é indiscutível a necessidade de obter dados da empresa que concretizem a modelação efetuada. A grande maioria dos dados acerca do modelo de negócio da *Farfetch* encontra-se disponível e em permanente atualização na sua base de dados, tendo sido extraídos os dados pretendidos através da linguagem SQL. É importante referir que a empresa possui, não uma, mas duas bases de dados, onde embora a estrutura de construção seja idêntica, o conteúdo é ligeiramente diferente, na medida em que existe informação numa que a outra não possui e vice-versa, para além de que são geridas por departamentos distintos. A base de dados designada por bidw é controlada por *Business Intelligence* ao passo que a base de dados denominada por bixpl é orquestrada pela equipa de operações.

A consulta SQL de dados foi um processo algo demoroso, precisamente pela necessidade de adquirir dados existentes nas duas bases de dados, o que obrigou a cruzar servidores e, consequentemente, tabelas distintas, com o cuidado de utilizar colunas únicas, evitando repetição de casos. No contexto do projeto, o método de recolha de dados foi baseado em dados históricos, mais especificamente em encomendas passadas de um determinado período. O período em causa poderia ser diário, semanal, mensal, sendo que o tempo de execução do programa SQL aumentaria com o aumento do período considerado, uma vez que a quantidade de dados a processar crescia consideravelmente. Estamos a falar em cerca de 6000 encomendas diárias, cerca de 5x mais semanalmente e 20x mais mensalmente. Este foi um fator levado em atenção. Como resultado, o conjunto de dados criados para fomentar este projeto não é mais do que uma matriz em que cada linha representa um pedido de um cliente.

A consulta realizada recupera os seguintes dados:

- **Encomendas:** identificadores das encomendas; identificadores dos artigos que compõem cada encomenda; identificador do *stock point* atribuído a cada artigo; identificador da loja onde foi comprado o artigo.
- **Localização:** país onde se localiza o *stock point* e respetivo código; país onde se localiza o cliente e respetivo código; continente ao qual pertence o país do cliente, para efeitos estatísticos.
- **Transporte:** serviço escolhido pelo cliente no momento da compra (*Express* ou *Standard*); transportadora que realizou o envio da encomenda; código postal do cliente; código postal do *stock point*; peso da encomenda.
- **Lucro Farfetch:** percentagem de comissão aplicada a cada produto consoante o *stock point* a que foi alocado; preço recomendado pela loja para a venda do artigo; *geomargin* ganha pela Farfetch com a venda do artigo.
- **Cliente:** velocidade de envio da encomenda entre os passos 1 e 5 (diferença entre a data de levantamento da encomenda e a data de criação); tempo em trânsito no passo 6.
- **Parceiros:** ano, número da semana, quantidade de *stock* inicial por produto, quantidade de vendas *online* por produto, quantidade de vendas *offline* por produto, *stock* no final da semana por produto, estação e categoria a que pertence o produto.

A *query* desenvolvida que permite alcançar os dados mencionados está descrita no anexo B.

Uma vez que os custos de transporte revelaram-se a métrica mais complexa, por dependerem de muitos fatores, a recolha de dados para o tratamento desta variável foi repartida no sentido em que foram elaboradas pequenas *queries* quer para a DHL, quer para a UPS, com o objetivo de extrair diretamente o custo expectável de envio em função do peso e da rota origem-destino. Estas pequenas *queries* estão também apresentadas no anexo B.

As bases de dados da Farfetch não foram a única fonte de dados, existindo um complemento realizado em *excel* através do preenchimento manual em parceria com a equipa de operações, no âmbito dos custos de transporte para a transportadora UPS, onde foram realizadas combinações de códigos postais origem-destino com o intuito de obter a zona a que pertence cada combinação. A ideia era cruzar a informação devolvida pelas *queries* com a informação introduzida manualmente no *excel*. Alguns excertos dos dados inseridos manualmente estão representados no anexo C. Excertos dos resultados devolvidos pelas *queries* concebidas podem também ser analisados no anexo B.

4.5 Simulação

As fases abordadas nas subsecções 4.2, 4.3 e 4.4 são os grandes *inputs* para a simulação da estrutura de otimização desenvolvida. A simulação engloba dois grandes processos: a construção do modelo e o método experimental. Os dois processos estão intimamente ligados, não fosse o

modelo o meio para atingir o fim, a experimentação. O modelo foi apresentado na subsecção 4.2 ao passo que a lógica da simulação será descrita mais à frente.

O principal objetivo da simulação é mostrar o comportamento do sistema perante a teoria e/ou hipótese construída, avaliando o seu impacto nas três perspetivas apresentadas nas subsecções anteriores. Para isso, foi projetado um modelo computacional em contextos idênticos aos da realidade que, no caso da *Farfetch*, entenda-se como sendo o modelo atualmente utilizado para seleção de parceiros. Assim, é possível antecipar impactos, corrigir erros ou falhas, antes de ser tomada uma decisão e da experiência se realizar efetivamente na realidade.

Para que a simulação possa funcionar como desejamos, é inevitável o consumo de dados, quer seja em tempo real ou de forma estática, traduzidos em variáveis controladas, daí a importância das etapas realizadas anteriormente, a modelação e a recolha de dados.

4.5.1 Etapas Iniciais

Ainda antes de progredir com a simulação, foi definido o ambiente de programação, assim como a linguagem a utilizar. No caso do ambiente de programação, era suficiente um editor de texto simples, disponível em diversos sistemas operativos, desde *macOS*, *Linux* e *Microsoft Windows* e que suportasse diversas linguagens de programação para que o leque de escolhas fosse diversificado. Verificados estes pressupostos, a escolha recaiu sobre um dos melhores editores de texto, o *Atom*, onde para além de ser um *software* livre, é bastante configurável, possuindo diversos pacotes, inclusive para a integração de consola interna, permitindo executar qualquer programa sem recorrer a mecanismos externos ao *Atom*. O sistema operativo adotado para a implementação foi o *Microsoft Windows*.

No que se refere à linguagem de programação, muitas foram as opções, no entanto, a eleição incidiu sobre a linguagem *python*. Trata-se de uma linguagem de alto nível, funcional, simples e intuitiva, com variados domínios de aplicação, bibliotecas e módulos reutilizáveis capazes de resolver inúmeras tarefas. A sua utilização está cada vez mais presente nas organizações.

Desde início, as potencialidades do *Atom* foram aproveitadas, começando precisamente pela instalação do pacote que assegura a existência de uma consola interna para testar o programa, conhecido por *Hydrogen*, revelando-se muito útil uma vez que assim apenas se justificava utilizar o editor de texto e nada mais.

O primeiro passo no avanço da simulação consistiu na leitura de dados em tempo real a partir de qualquer uma das duas bases de dados da empresa. Para isso, foi realizada a conexão a ambas de igual forma, começando por importar as definições do módulo *pyodbc*, um módulo que permite ligar aplicações *python* a fontes de dados através de um *driver* ODBC (Conetividade aberta de base de dados). Este tipo de *driver* usa a interface ODBC da *Microsoft*, permitindo que as aplicações acedam a sistemas de gestão de base de dados usando como padrão a linguagem SQL.

Atualmente, existem muitos *drivers* ODBC de servidores SQL escritos e distribuídos pela *Microsoft*, entre eles aquele que figura no presente na *Farfetch*: *SQL Server Native Client 11.0*. A sequência de conexão para este *driver* e para a maioria dos restantes é idêntica e foi realizada usando o método *connect()* do objeto *connection*. Este método requer os seguintes aspetos:

- **Driver:** identifica o *driver* que se deseja usar para se conectar à base de dados. No caso da *Farfetch* estamos perante o SQL *server*, em particular o SQL *Server Native Client* 11.0, como já referido acima.
- **Servidor:** identifica o servidor em que o SQL *server* está a ser executado. O servidor utilizado foi o *bidw.farfetch.net* ou *bixpl.farfetch.net*, consoante a base de dados a que se pretendia aceder.
- **Base de dados:** identifica o nome da base de dados. *bi_dw* ou *bixpl* são os nomes das bases de dados da *Farfetch*.

Uma vez que as operações numa base de dados relacional são feitas a partir de um conjunto de linhas ou resultados, e tendo em conta que interessa trabalhar linha a linha, recorreu-se àquilo a que se designa de um cursor. Um cursor é um objeto da base de dados que percorre um conjunto de resultados, facilitando a sua manipulação sequencial linha a linha. Pode ser visto como um apontador para uma linha de entre um conjunto de linhas.

O objeto cursor foi instanciado a partir do método *connection.cursor()*, estando pronto a operar numa base de dados. Utilizando o método *execute()*, essa operação é preparada e executada (consulta ou comando).

Quer a ligação à base de dados, quer a criação de um objeto cursor para processamento de dados se referem a tarefas que foram criadas dentro de duas funções um pouco distintas: *read_query_file* e *read_query_file_com()*. A função *read_query_file_com()*, realizada em parceria com a equipa de operações, recebe como parâmetro ficheiros SQL cujas *querys* criam novas tabelas na base de dados com a informação pretendida. Esta função revelou-se necessária para a métrica dos custos de transporte, na medida em que a informação existente estava incompleta.

A função *read_query_file()* recebe como parâmetro o nome do ficheiro SQL com a *query* que devolve a consulta desejada, o que significa que sempre que pretendermos consultar dados em tempo real, invocamos a função com o respetivo ficheiro. Com a utilização do método *cursor.fetchall()*, todas as linhas do resultado da consulta foram recuperadas, retornando uma lista de tuplas, que tem a particularidade de não deixar alterar os seus valores. Caso não seja devolvido nenhum resultado, o método retorna uma lista vazia.

Com o intuito de facilitar o acesso e a manipulação de dados ao longo do programa, sempre que era feita uma consulta, era criada uma estrutura de dados bidimensional designada por *data frame*, que se caracteriza pela sua forma tabular com eixos bem definidos (linhas e colunas). Os principais parâmetros de uma *data frame* são os identificadores das linhas e colunas, assim como os dados que a estrutura deve ter. Caso se omita o parâmetro referente às linhas, a estrutura assume um padrão de índices para um determinado intervalo (0, 1, 2, ..., n), em que n é o número de linhas retornadas.

No que diz respeito às colunas, foi usado o método *cursor.description()* que permite obter, entre outros detalhes, o nome das colunas. Este tipo de estrutura encontra-se disponível num dos pacotes mais populares para *python*, o *pandas*, que oferece diversas estruturas de dados poderosas,

flexíveis e expressivas. Este pacote foi, por isso, importado para o programa, para desse modo ser possível declarar uma *data frame*.

O método utilizado para aceder às várias posições de uma *data frame* foi através do *pandas.dataframe.loc*. *.loc[]* acede a um grupo de linhas e colunas por rótulo, ou se quisermos, pelo seu nome. Um pequeno exemplo: *results.loc[1]['profit']*. A *data frame* "result" acede ao valor que está na linha com índice 1 e na coluna denominada por "profit".

4.5.2 Lógica

Implementados e decididos todos os requisitos necessários ao funcionamento da simulação, a sua lógica pode ser dividida em 4 grandes etapas. A primeira etapa incide na declaração, sobretudo de variáveis. Em primeiro lugar, foi essencial concretizar a hipótese modelada na secção 4.3 do papel para o programa, já que a simulação pretendia quantificar o seu efeito. Para cada uma das três perspetivas em análise foi definida uma ponderação inicial, ou se preferirmos, uma importância relativa que cada uma teria na seleção de parceiros. Como estamos perante um ambiente de simulação, a decisão tomada foi a de que o peso inicial de cada perspetiva seria igual, tendo sido atribuído 0,333 (33,3%) a cada uma. Estes valores são controláveis, no sentido em que podem ser alterados a qualquer momento caso se pretendam analisar vários cenários. Deste modo, o polinónimo apresentado na equação 4.14 foi inserido na simulação da seguinte forma:

$$\begin{aligned} \text{Hipótese} = & \frac{1}{3} \times (\text{Comissão} + \text{Geomargin} - \text{Custos de transporte}) + \\ & \frac{1}{3} \times (\text{Peso de vendas online} \times \text{Sell Through}) + \frac{1}{3} \times (\text{Tempo de entrega bruto}) \end{aligned} \quad (4.14)$$

(0.333; 0.333; 0.333) foi considerada a solução inicial para efeitos de simulação, tendo sido definidas três variáveis, cada uma delas assumindo o papel de uma perspetiva.

A próxima declaração, realizada em parceria com a equipa de operações, determinou que *targets* seriam considerados para cada perspetiva. Por *targets*, neste contexto, entenda-se como valores ideais ou desejados que se pretende que sejam alcançados para o lucro da *Farfetch*, para a satisfação dos clientes e para o crescimento dos parceiros. Os *targets* definidos foram de \$100 para lucro, 5 dias para clientes e 70% para parceiros. É importante realçar que estamos a falar de valores com grandezas distintas uma vez que as próprias métricas assim o exigem. Basta perceber que o lucro é expresso em dólares, o prazo de entrega para clientes em dias e as métricas referentes a parceiros em percentagem.

A segunda grande etapa da simulação lê e percorre todos os resultados devolvidos pela query acerca do histórico de encomendas, descrita no anexo B, onde cada linha representa um pedido efetuado por um cliente no dia 15 de Abril de 2018. Para cada uma dessas encomendas, e estamos a falar em cerca de 3000 encomendas referentes a este dia, o programa verifica o tipo de serviço selecionado pelo cliente (*Express* ou *Standard*), assim como a transportadora que efetuou a entrega. Esta verificação é efetuada desde logo uma vez que, como referido anteriormente, os custos de transporte são diferentes consoante o serviço e, sobretudo, a transportadora. Deste modo, a

simulação agrupa as condições de verificação em dois grandes grupos, cada um subdividido em dois subgrupos: Um primeiro grupo para a DHL, subdividido em serviço *Express* e *Standard*. Para o serviço *Express* estão associadas as transportadoras *DHL Express*, *DHL Express Brazil* e *DHL Express China*, enquanto que para o serviço *Standard*, temos a *DHL Ground* e a *DHL Economy*.

O segundo grupo, destinado às encomendas enviadas pela UPS, a condição de verificação ganha um acréscimo face ao caso da DHL. Para além de ser analisado o tipo de serviço e a transportadora em detalhe, *UPS Second Day Air* para serviço *Express* e *UPS Standard* ou *UPS Ground* para serviço *Standard*, o programa averigua se a origem e o destino da encomenda é os Estados Unidos da América. Isto porque, trata-se de um país onde a UPS atua em grande escala.

Quando uma encomenda entra na condição relativa à transportadora DHL, quer seja *Express*, ou *Standard*, a simulação compara o código do país do cliente que efetuou esse pedido, o código do país do *stock point* que o expediu e o peso da encomenda, percorrendo os resultados retornados pela *query* acerca dos custos de transporte para a DHL, até ser encontrada a combinação de códigos origem-destino com o peso pretendido associado. A combinação destes três aspetos tem um custo de envio, que não é mais do que o custo de transporte que essa encomenda trouxe para a empresa.

Caso se trate de uma encomenda cuja transportadora atribuída foi a UPS, quer o serviço tenha sido *Express* ou *Standard*, o programa extrai os três primeiros caracteres ou dígitos do código postal do cliente e do *stock point*, percorrendo de seguida o documento *excel* desenvolvido para a transportadora em questão onde estão as combinações de códigos postais origem-destino com a respetiva zona. No momento em que é encontrada a combinação para a encomenda, a simulação guarda a zona a que pertence numa variável auxiliar uma vez que é necessária para decifrar o custo de envio.

Mais uma vez, o programa lê e percorre dados inseridos em *excel*, neste caso, um conjunto de zonas e respetivos preços, até ser encontrada a zona em questão, altura em que o custo associado a essa zona é, também ele, guardado numa variável auxiliar para uso futuro.

Simultaneamente, a simulação busca para cada artigo de uma encomenda o preço recomendado pela loja para venda, aplicando sobre este a percentagem de comissão definida no contrato, assim como a *geomargin* ganha pela *Farfetch* com a venda de cada produto. Recolhidos que estão todos os dados relacionados com o lucro, custos de transporte, *geomargin* e comissões, o programa calcula o valor para a perspetiva, não esquecendo de aplicar a ponderação inicialmente definida, 0,33, tal como evidencia a equação 4.14. Logo de seguida, vem a perspetiva de clientes, para a qual a simulação recolhe o valor, em dias, do tempo que a encomenda demorou a chegar ao cliente, o chamado tempo de entrega. O cálculo intermédio da métrica, a soma do *speed of sending* com o *time in transit* é realizado pela própria *query* e, portanto, a simulação mostra apenas o valor final. A este valor aplica também a ponderação inicial de 0,33, resultando no valor final para esta perspetiva.

Por último, o programa percorre linha a linha os resultados da *query* acerca de *Sell Through*, cujo um excerto está presente no anexo B, comparando os identificadores de produtos. Como as encomendas simuladas dizem respeito ao dia 15 de Abril de 2018, a análise de parceiros foi realizada pela simulação para a semana de 15 a 22 de Abril. De um lado os produtos que compõe

cada encomenda efetuada nesse dia, do outro todos os artigos que estavam à venda nessa semana, o que implica que os artigos comprados pelo cliente nesse dia coincidam com parte dos produtos que estavam para venda ao longo da semana correspondente. Para cada *item* de uma encomenda, o programa calculou o seu *Sell Through* de acordo com a equação 4.6 assim como o peso de vendas *online* que esse *item* alcançou, como sugere a equação 4.7. A simulação utilizou estes dois valores para calcular, de seguida, a percentagem de vendas *online* de que esse artigo foi alvo perante o seu *stock* inicial e excluindo as vendas ocorridas na loja, de acordo com a equação 4.8. Tal como aconteceu com as anteriores perspetivas, o valor final para a perspetiva de parceiros foi também sujeito à ponderação inicial de 33,3%.

As perspetivas e respetivas métricas determinadas pela simulação foram apresentadas a partir de um *data frame* cujas linhas representam os índices das encomendas ao passo que as colunas foram definidas consoante aquilo que interessava obter. A simulação exibe o *Portal Order id* da encomenda, o identificador do produto, o continente de destino, o identificador da loja, o identificador do *stock point*, a estação, a categoria e a família a que pertence cada produto para efeitos estatísticos e as três últimas colunas, cada uma mostrando os valores referentes a cada perspetiva: lucro, clientes e parceiros, como está patente no anexo E.

A terceira etapa da simulação está relacionada com a análise dos *targets*, no sentido em que o programa, para cada encomenda, calcula a diferença existente entre valores obtidos em cada perspetiva e o respetivo valor desejado. A diferença obtida foi também guardada a partir de um *data frame*, desta vez com apenas três colunas, cada uma representando, novamente, uma perspetiva.

A última etapa, crucial para a restante estrutura de otimização como veremos mais à frente, consiste na normalização de todos os dados obtidos. Para isso, o programa começa por determinar, encomenda a encomenda, o valor máximo e mínimo registado em cada perspetiva, através das funções `max()` e `min()` que a linguagem *python* disponibiliza, guardando esses valores em variáveis auxiliares. Os dados foram normalizados através da equação 4.15:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4.15)$$

onde X representa o valor obtido da simulação, X_{min} o valor obtido através da função `min()`, X_{max} obtido pela função `max()` e X' o respetivo valor normalizado na escala unitária.

Para o caso dos valores relativos a cada um dos *targets*, o processo de normalização foi um pouco distinto uma vez tratar-se de valores únicos. Para cada *target* definido, o programa verificou se era superior ao valor máximo encontrado para a respetiva perspetiva, e caso se confirmasse, o *target* era normalizado para o valor unitário. Caso contrário, em que o *target* se revelava inferior ao valor máximo encontrado, foi aplicada a equação 4.15 para normalizar esse valor.

O código correspondente a toda a lógica de simulação está presente no anexo D.

4.5.3 Resultados

Assim que a simulação terminou de correr, os resultados foram analisados sob diferentes pontos de vista. Começando pela perspectiva de lucro para a *Farfetch*, a simulação revelou a existência de algumas encomendas com valores negativos, o que significa que estes pedidos, com determinados *stock points* atribuídos, assim como transportadoras alocadas não trouxeram benefício para a empresa. Tendo em conta a forma como a perspectiva foi modelada, a explicação para tal facto é a de que os custos de envio revelaram-se superiores comparativamente com aquilo que a empresa ganhou em comissões e *geomargin* através da venda dos artigos dessas encomendas.

O valor máximo de lucro registado pela simulação para o dia em questão foi de \$1963.103, referente ao artigo 3591617 da encomenda 2E2TV2, cuja origem foi o *stock point* 1271 em Itália e o destino, o cliente na Alemanha. O valor mínimo registado, conotado como perda, foi de -\$38.71, respeitante à venda do artigo 3276921 da encomenda ZJZPQQ.

Relativamente aos resultados obtidos para a perspectiva de clientes, e partindo da única métrica usada, o prazo de entrega, verificou-se que a entrega mais rápida foi efetuada pela DHL *Express*, tendo demorado 1,4 dias, mais especificamente, 1 dia e 10 horas aproximadamente. Já a entrega mais demorada, ocorreu entre Portugal e a Dinamarca, efetuada pela DHL *Express*, ao longo de 39 dias e 17 horas, aproximadamente, o que revela alguma disparidade de valores como consequência de diversos fatores, entre eles a distância entre a origem e o destino, a transportadora, entre outros.

No que se refere às vendas realizadas pelos parceiros na semana de 15 de Abril para os artigos correspondentes às 2877 encomendas dos clientes, 330 produtos foram apenas vendidos através do portal da *Farfetch*, registando uma percentagem de vendas *online* de 100%. No entanto, e desequilibrando um pouco a balança, 815 foram vendidos apenas em loja e, como tal, registaram um percentagem de vendas no portal de 0%.

A segunda parte da análise focou-se em alguns aspetos estatísticos, nomeadamente em termos de mercados com quem a *Farfetch* diariamente interage. A simulação partiu de 2877 encomendas, e, recorrendo a um filtro por continente, foi verificado não terem sido realizadas quaisquer encomendas a partir do continente Africano e Índia (0%), 1% (29 encomendas) tiveram como destino a América Latina e *Caribbean*, 3,7% (106 pedidos) foram feitos a partir do Médio Oriente, 24% das 2877 encomendas entregues nos Estados Unidos da América, 31% (877 encomendas) concluídas para a Ásia e 41% entregues por toda a Europa. Estes valores, apresentados através de um gráfico circular, como pode ser visto no anexo F revelam que, de facto, os mercados com maior expressão no modelo de negócio da *Farfetch* são, sem dúvida, o mercado europeu, asiático e norte-americano.

No que se refere ao alcance dos valores desejáveis e definidos no início da simulação, os *targets*, a sua avaliação seguiu, igualmente, a permissa anterior. É importante realçar que nesta análise as diferenças face aos *targets* positivas não foram penalizadas, tem sido calculado o $\max(0, \text{target} - \text{resultado})$ para cada um dos critérios. Para a perspectiva de lucro, e apesar de existirem mercados com uma maior percentagem de pedidos diariamente, o lucro médio para o dia considerado apresentou valores bastante idênticos de mercado para mercado. Por exemplo, e como mostra o

primeiro gráfico de barras presente no anexo F, o lucro médio obtido em encomendas enviadas para a América Latina e *Caribbean* foi de \$18,94, muito próximo dos \$19,00 obtidos a partir de pedidos cujo destino foi a Europa. Estes valores são a prova de que, apesar de serem recebidos poucos pedidos de um determinado mercado, pode acontecer que esses pedidos gerem mais receita, indiciando que aquele mercado merece ser potencializado. Todos os mercados apresentaram lucros médios entre os \$18,94 e os \$19,06, com exceção de *Africa and India*, valores abaixo do montante desejável para esta perspetiva, fixada nos \$33. O intervalo de erro variou entre os [13,94; 33] com uma amplitude de 19,06. É importante referir que estes são valores com a ponderação de 33,3% já aplicada.

Em relação à satisfação dos clientes, a situação não é muito distinta, na medida em que o tempo médio que uma encomenda demorou a chegar ao destino praticamente não variou consoante o mercado a que nos referimos. Estamos a falar de valores entre os 1,3028 e os 1,3048 dias. Tal como se apurou na perspetiva de lucro, no caso dos clientes os valores obtidos encontram-se também abaixo do *target* desejado para esta perspetiva, situado nos 1,65 dias. Isto significa que os clientes estão a esperar menos tempo pelos seus produtos do que aquilo que a *Farfetch* considera como ideal. O intervalo de erro esteve compreendido entre os [0,3452; 1,65], o que corresponde a uma amplitude de 1,3048.

Por último e relativamente ao crescimento dos parceiros foi realizada uma avaliação das suas vendas *online* por família de produto e por estação. Quer num caso, quer no outro, os valores não sofreram grandes oscilações, o que revela uma grande diversidade de compra no portal. É importante destacar que, claramente, os artigos das estações mais recentes de primavera-verão e outono-inverno apresentam uma procura maior e que as roupas, o calçado e as mochilas são as famílias de produtos mais requisitadas. Em termos de proximidade com a percentagem de vendas *online* pretendida pela empresa, 70% traduzida em 23,1% devido à ponderação de 33,3%, os valores obtidos, tal como nos casos anteriores, encontram-se abaixo dessa meta, com a exceção de artigos da estação primavera-verão de 2010 que registou uma média de vendas *online* no período considerado superior ao desejável, tal como está patente no anexo F. Os intervalos de erro face ao *target* fixado posicionaram-se em [0; 19,433] para as vendas médias por estação e [10,870; 23,1] para as vendas médias por família de produto, com amplitudes respetivas de 19,433 e 12,23.

4.6 Otimização

A última grande etapa que compõe a estrutura para a cadeia de abastecimento da *Farfetch* surge pela necessidade de ajustar e controlar a hipótese definida na secção 4.3 e complementada na subsecção 4.5.2 com a atribuição de ponderações iniciais a cada perspetiva do sistema. A simulação permitiu perceber e avaliar o impacto que a hipótese causou nas três perspetivas delineadas ao nível de cada um dos *targets*, tal como foi explicado na subsecção 4.5.3, para agora, a estrutura iniciar uma análise de regressão com o intuito de se perceber o comportamento da variável de saída dependente, o *target* definido para cada critério, face às mudanças das variáveis de entrada independentes, os valores calculados para cada critério através da simulação. Pretende-se alcançar

uma relação entre as três perspetivas. Por relação entende-se encontrar as melhores ponderações para cada perspetiva de forma a que os valores ideais sejam atingidos e, dessa forma, a diferença entre os resultados obtidos na simulação e os *targets* seja mínima.

A abordagem seguida para esta última fase da estrutura foi a construção de uma rede neuronal *feedforward* multi-camada com método de treino *backpropagation*, recorrendo igualmente à linguagem *python* e ao editor de texto *Atom*. A escolha recaiu sobre esta arquitetura e sobre este método uma vez que aquilo que se pretendia era uma atualização e mudança iterativa dos pesos atribuídos a cada perspetiva na hipótese definida, para que a rede neuronal pudesse aprender a mapear corretamente as entradas nas saídas, levando a que o erro à saída fosse convergindo para zero, com a aproximação aos valores desejados.

Com o objetivo de facilitar e simplificar a sua implementação foi criada uma classe para a rede neuronal, permitindo assim o agrupamento de dados e funcionalidades que, de certa forma, possuem propriedades ou atributos em comum. Criar uma nova classe implica a criação de um novo tipo de objeto, permitindo que novas instâncias desse tipo fossem definidas. As declarações realizadas dentro da classe foram um conjunto de funções ou métodos com os respetivos argumentos, essenciais para a concretização da rede e que serão detalhadas à frente.

4.6.1 Inicialização

O primeiro passo no desenvolvimento da rede neuronal foi a sua inicialização. Para este processo foram utilizados dois métodos chave: *self* e *__init__()*. Para o sucesso de ambos muito contribuiu a classe definida uma vez que permitiu a criação de um objeto dessa mesma classe, onde foram indicados os argumentos que pretendíamos passar. Ora, estamos a falar do número de nós, ou neurónios, que cada camada da rede neuronal iria possuir. Tendo em conta toda a estrutura desenvolvida até esta fase, para a camada de entrada ficou estabelecido que seria composta por três nós, já que pretendíamos otimizar três grandes perspetivas, logo cada neurónio estaria associado a uma perspetiva diferente. O neurónio 1 seria para a perspetiva de lucro, o neurónio 2 para parceiros e o neurónio 3 para clientes. Adicionalmente, foi incluído na camada de entrada um nó extra para aquilo a que se designa por *bias*, algo indispensável para uma rede neuronal de treino com o algoritmo *backpropagation*. Este nó terá sempre o valor de entrada 1 e sem ele a rede não aprende.

A camada intermédia, conhecida por camada oculta, foi definida com apenas um neurónio, o neurónio 4, quer por razões de menor complexidade, quer pela natureza do problema, já que não se revelou necessário optar por uma camada mais extensa. E por fim, a camada de saída, que tal como a camada de entrada foi inicializada com três nós, isto porque na saída de cada nó era expectável que os *targets* de cada perspetiva fossem alcançados. O neurónio 5 estaria associado ao *target* de lucro, o neurónio 6 ao *target* de parceiros e o neurónio 7 ao de clientes.

O número de nós de cada camada foi passado pelo objeto ao método *__init__()*, definido com o intuito de inicializar os atributos da classe. A ligação entre os atributos e os argumentos foi feita através do método *self* que não é mais do que uma instância da classe. Vejamos um exemplo: a inicialização do número de nós na camada de entrada resultou da seguinte forma: *self.ni = NI*,

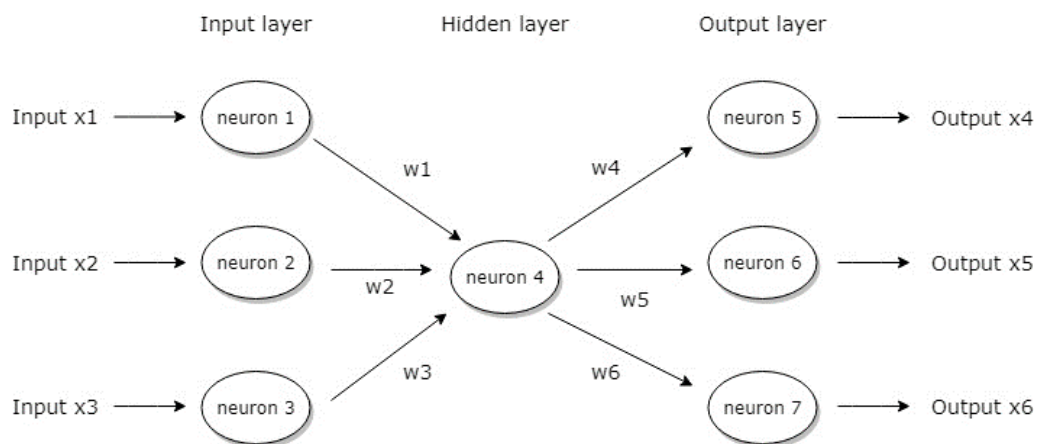


Figura 4.3: Configuração da rede neuronal projetada

onde *ni* é o atributo e *NI* o argumento passado pelo objeto que, neste caso, foi 3. A configuração da rede neuronal implementada está expressa na Figura 4.3.

Para além da definição da estrutura das camadas da rede neuronal, o método `__init()` conheceu outras inicializações. Foram criados três vetores, um para cada camada, com o intuito de inicializar cada nó, o que implica que cada vetor possua um número de posições igual ao número de neurónios que a camada respetiva tem. Seguiu-se o tratamento dos pesos entre as várias camadas, representados na Figura 4.3 por *w1*, *w2*, *w3*, *w4*, *w5* e *w6*. Os pesos são valores que representam o grau de importância que determinada entrada possui em relação a determinado neurónio. Com a intenção de guardar os seus valores, foram criadas duas matrizes com a mesma dimensão, 3x1. Uma das matrizes guarda os três pesos que fazem a conexão entre a camada de entrada e a camada oculta (*w1*, *w2* e *w3*), enquanto que a segunda matriz guarda os pesos que fazem a conexão entre a camada oculta e a camada de saída (*w4*, *w5* e *w6*). As matrizes foram originadas através da função `makeMatrix()` que recebe como parâmetros o número de linhas e colunas.

A inicialização das matrizes foi realizada através da função `randomizeMatrix()` que recebeu como parâmetros a matriz criada, ainda vazia, assim como o intervalo de valores a partir dos quais os pesos seriam gerados aleatoriamente. Neste caso, os valores para os pesos *w1*, *w2*, *w3*, *w4*, *w5* e *w6* foram gerados para o intervalo de 0 a 1.

O último passo na inicialização da rede neuronal consistiu na criação de duas outras matrizes, igualmente através da função `MakeMatrix()`, de igual dimensão às anteriores, mas com um objetivo diferente, guardar as alterações dos pesos para o momento.

O código referente à inicialização da rede neuronal pode ser analisado no anexo G.

4.6.2 Propagação Direta

Concluída a construção e inicialização da rede neuronal, a mesma encontra-se preparada para receber dados. No conjunto de dados, os dados de entrada *X* (*x1*, *x2* e *x3*) representam os valores obtidos para as três perspetivas através da simulação. Nesta fase, as entradas foram propagadas

até serem obtidos valores nas saídas definidas, procedimento conhecido por propagação direta. Assim, foi criado um novo método chamado de *runNN()* que recebe como parâmetro os dados de entrada. Caso o número de entradas seja diferente do inicialmente definido na inicialização da rede, o programa exibe um erro a reportar o número incorreto de entradas, não prosseguindo. Caso contrário, os dados são guardados num vetor 3x1 criado aquando da inicialização para a camada de entrada para utilização futura.

Segue-se a determinação do valor de entrada para cada neurónio da camada oculta. Para isso, o programa começa por inicializar uma variável auxiliar a zero, *sum*, percorrendo depois cada neurónio na camada de entrada com intuito de aceder ao valor de cada um. Uma vez que a rede projetada possui apenas um neurónio na camada oculta, como mostra a Figura 4.3, o seu valor de entrada foi calculado de acordo com a equação 4.16,

$$neuron_4 = x1 \times w1 + x2 \times w2 + x3 \times w3 \quad (4.16)$$

onde cada elemento do vetor da camada de entrada ($x1$, $x2$ e $x3$) necessita de ser multiplicado pelo peso correspondente gerado aleatoriamente que faz a conexão entre o nó da camada de entrada e nó da camada oculta. Porém, o resultado obtido e guardado na variável *sum* não é o valor final para a camada oculta. Como em qualquer nó, qualquer que seja o contexto, existe uma entrada e uma saída, o que significa que o próximo passo foi o cálculo da saída para o nó da camada oculta. Este cálculo foi efetuado recorrendo àquilo a que se designa por função de ativação. Existem vários tipos de funções de ativação e a sua utilização em redes neuronais é muito importante, no sentido em que introduzem uma componente de não linearidade, aumentando a capacidade de aprendizagem da rede e facilitando a alteração dos pesos no futuro. A função de ativação escolhida foi a função *sigmoid* ou logística já que permite modelar comportamentos cujos valores variem entre 0 e 1, o que vai de encontro com a normalização dos dados da simulação realizada e abordada na subsecção 4.5.2. A forma gráfica da função *sigmoid*, Figura 4.4, e a sua representação matemática na equação 4.17 traduzem isso mesmo.

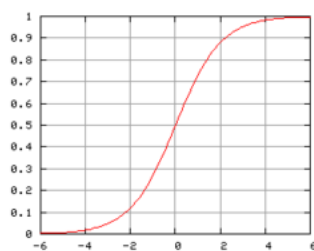


Figura 4.4: Função sigmoid

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.17)$$

A função *sigmoid()* recebe como parâmetro a variável *sum* com o valor de *neuron₄*, retornando o

resultado final para a saída do neurónio da camada oculta dado pela equação 4.18, um valor entre 0 e 1 que fica guardado no vetor 1x1 criado durante a inicialização da rede.

$$f(neuron_4) = \frac{1}{1 + e^{-neuron_4}} \quad (4.18)$$

O processo repete-se para a camada de saída. O programa acede à camada oculta e utiliza o seu valor final para calcular os valores da camada de saída. Estes valores foram guardados, tal como no caso anterior, numa variável auxiliar. Tendo a camada de saída três neurónios, os seus valores foram calculados de acordo com as equações 4.19, 4.20 e 4.21,

$$neuron_5 = f(neuron_4) \times w_4 \quad (4.19)$$

$$neuron_6 = f(neuron_4) \times w_5 \quad (4.20)$$

$$neuron_7 = f(neuron_4) \times w_6 \quad (4.21)$$

onde w_4 , w_5 e w_6 são os pesos gerados aleatoriamente que fazem a conexão entre o nó da camada oculta e cada um dos nós da camada de saída. Tal como aconteceu com o neurónio da camada oculta, o valor final de cada neurónio da camada de saída foi obtido através da função *sigmoid()*, equação 4.17, e guardados no vetor 3x1 inicializado logo no arranque da rede neuronal. Estamos a falar dos valores $f(neuron_5)$, $f(neuron_6)$ e $f(neuron_7)$. O processo de propagação direta termina neste ponto com a obtenção de resultados nas saídas face aos pesos gerados e o seu código está presente no anexo G.

4.6.3 Retropropagação

Perante os pesos gerados aleatoriamente e os dados injetados na entrada, a rede neuronal é capaz de mostrar na saída o resultado obtido. Esse resultado pode, no entanto, estar perto ou longe do valor que seria desejável. É neste período que entra em ação o processo de retropropagação cujo principal objetivo passa pelo treino da rede, garantindo que a saída real ficará próxima da saída alvo, minimizando, assim, o erro de cada neurónio de saída e da rede como um todo. O meio para atingir este fim inclui realizar uma mudança incremental nos pesos e a forma como essa mudança foi feita será descrita a seguir.

Seguindo o mesmo raciocínio, foi criado um novo método, denominado de *backPropagate()*, que recebeu como parâmetros os valores pretendidos à saída e a taxa de aprendizagem. O primeiro passo na concretização do método foi a inicialização de um vetor com a dimensão do número de saídas, 3, para utilização futura.

Para descobrirmos que direção seguir na alteração dos pesos foi necessário determinar qual a taxa instantânea de mudança do erro em relação aos pesos. Por outras palavras, foi utilizada a derivada para entendermos de que modo os pesos afetam as entradas da rede. Este método é conhecido por gradiente descendente e permitiu determinar o modo de alteração dos pesos que levaria a valores de saída mais precisos.

Como primeiro passo, o programa acedeu ao valor resultante do processo de propagação direta para cada nó de saída e calculou o erro acerca do quão distante se encontrava cada um desses valores do valor esperado, tal como mostra a equação 4.22,

$$error_{out} = target - f(neuron_i), i \in [5,7] \quad (4.22)$$

onde i representa cada nó da camada de saída. Estes erros são definidos como atributos de cada nó de saída e não a taxa final de que necessitamos. Sobre cada um destes erros foi aplicada a derivada da função de ativação *sigmoid*, traduzida pela função *dsigmoid()* que recebe como parâmetro o valor final de saída de cada nó, sum_i . O cálculo por detrás da função *dsigmoid()* está demonstrado na equação 4.23 e o cálculo sobre o erro, na equação 4.24.

$$f'(neuron_i) = f(neuron_i) \times (1 - f(neuron_i)), i \in [5,7] \quad (4.23)$$

$$output_delta_i = error_{out} \times f'(neuron_i), i \in [5,7] \quad (4.24)$$

A derivada final de cada nó de saída foi obtida multiplicando o $output_delta_i$ respetivo pela saída final do nó da camada oculta, como demonstra a equação 4.25.

$$change_i = output_delta_i \times neuron_i, i \in [5,7] \quad (4.25)$$

O $output_delta_i$ respeitante ao erro de cada nó da camada de saída foi usado para perceber o quanto a camada oculta contribuiu para o erro de saída. Para isso, o programa começou por calcular o erro à saída do único neurónio da camada oculta de acordo com a equação 4.26,

$$error_{hidden} = \sum_{i=4}^6 output_delta_i \times w_j, j \in [4,6] \quad (4.26)$$

onde w_j representa o peso que faz a conexão entre o neurónio da camada oculta e o neurónio i da camada de saída. Tal como aconteceu para os neurónios da camada de saída, também para o da camada oculta foi aplicada a derivada da função sigmoid sobre o erro obtido, equação 4.27.

$$hidden_delta = error_{hidden} \times f'(neuron_4) \quad (4.27)$$

A derivada final para o nó da camada oculta foi determinada de igual forma, como exemplifica a

equação 4.28.

$$change_4 = hidden_{delta} \times neuron_4 \quad (4.28)$$

Conhecida a taxa final de mudança do erro quer para a camada de saída, quer para a camada oculta, designada de *change*, a rede neuronal está pronta para o ajuste de pesos. Este ajuste ocorreu de forma semelhante quer para os pesos da camada de saída (w_4, w_5, w_6), quer para os pesos da camada de oculta (w_1, w_2, w_3). O ajuste de w_4, w_5 e w_6 seguiu a equação 4.29 e o ajuste de w_1, w_2 e w_3 a equação 4.30,

$$w_j = w_j + N \times change_i + M \times changeBefore_i, j \in [4,6], i \in [5,7] \quad (4.29)$$

$$w_j = w_j + N \times change_4 + M \times changeBefore_4, j \in [1,3] \quad (4.30)$$

onde w_j do lado direito das equações representa o peso gerado na iteração anterior, *changeBefore_i* e *changeBefore₄* a taxa final para o nó de saída i e a taxa final para o nó da camada oculta, respetivamente, na iteração anterior, N a taxa de aprendizagem e M a taxa de impulso. Nesta fase, o programa guarda as taxas finais calculadas para cada um dos nós de saída, *change_i* e para o nó da camada oculta, *change₄*, em vetores definidos para o efeito aquando da inicialização da rede uma vez tratarem-se de valores que foram utilizados a cada iteração.

O processo de retropropagação termina com o cálculo do erro total que mostra o quão longe esteve a rede das saídas desejadas. O erro total foi calculado de acordo com a equação 4.31, resultando no somatório do erro de perda para cada neurónio de saída.

$$total_error = total_error + (0.5 \times (target_i - neuron_i)^2), i \in [5,7] \quad (4.31)$$

A parte de código respeitante ao processo de retropropagação está presente no anexo G.

4.6.4 Treino

A lógica de funcionamento da rede neuronal é baseada no processo de treino de dados que engloba o processo de propagação direta e retropropagação. Para que a fase de treino pudesse ter início impunha-se a criação de uma função que invocasse pela ordem correta todos os métodos criados responsáveis pela execução da rede neuronal. Como em qualquer programa, essa função foi nomeada de *main()*. Nesta função foram tomados quatro passos imprescindíveis. O primeiro deles, a definição dos dados para treino através de uma matriz 2x3 implementada como uma lista de duas listas de comprimento 3. A primeira posição da lista ficou alocada para os dados de entrada e a segunda posição para os *targets*. O dados de entrada dizem respeito aos valores obtidos

da simulação para as três perspetivas em análise (lucro, clientes e parceiros) e os *targets*, os três valores definidos como ideais para cada perspetiva.

O segundo passo declarado na função *main()* foi o número de neurónios que se pretendiam em cada camada da rede, cujo processo de inicialização está descrito na subsecção 4.6.1. Definidos os dados e inicializada a rede neuronal, vem o terceiro passo responsável pela invocação do método *train()* que, como o próprio nome indica, permitirá o treino da rede. A função designada por *train()* recebeu como parâmetros os dados definidos para treino, o número de iterações a executar, a taxa de aprendizagem *N* e a taxa de impulso *M*. A taxa de aprendizagem avalia a contribuição relativa de cada peso no erro total, o que significa que a escolha do seu valor tem bastante importância. Se for um valor pequeno, a progressão de aprendizagem da rede será mais lenta enquanto que um valor mais elevado prossegue de forma mais rápida, podendo produzir oscilações entre soluções inadequadas. Perante estes dois cenários, ficou decidido que a taxa *N* teria o valor de 0.3. Quanto à taxa de impulso, revela-se útil para acelerar a convergência e evitar mínimos locais no sentido de estabilizar as alterações dos pesos. A taxa *M* foi definida com o valor de 0.4.

O programa identificou o conjunto de dados, fazendo a separação entre dados de entrada e *targets* a atingir em cada saída, estando pronto para iniciar o processo de treino. Os processos de propagação direta e retropropagação foram desencadeados. O primeiro método a ser chamado foi o *runNN()*, responsável pela execução e desempenho do processo de propagação direta explicado em detalhe na subsecção 4.6.2. Terminado o processo de propagação direta, o programa invoca o método *backPropagate()* que propaga o erro para trás através da rede neuronal, como detalhado na subsecção 4.6.3.

O método de treino imprime ainda o erro total a cada iteração cometido pela rede, cujo cálculo é efetuado no final da retropropagação, como explicado na equação 4.31. O método *train()* termina com a invocação da função *test()*, criada para imprimir as entradas, as saídas obtidas e os *targets* para cada linha de treino, terminadas as iterações definidas, permitindo assim analisar se as saídas reais se vão aproximando, ou não, das saídas desejadas. O método *test()* acede às saídas reais chamando o método *runNN()* que devolve essas mesmas saídas.

Com o intuito de perceber os pesos finais resultantes do processo de treino, quer para a camada intermédia, quer para a camada de saída, o quarto passo da função *main()* invoca o método *weights()* que imprime esses mesmos pesos, acedendo para isso aos vetores onde foram sendo guardados iteração a iteração.

A rede neuronal entra em ação imediatamente após a simulação descrita na secção 4.5 terminar de correr, sendo que o número de iterações impostas equivale ao número de encomendas simuladas. No caso, foram simuladas 2877 encomendas, o que significa que a rede neuronal terá 2877 iterações. Cada linha de dados, ou cada encomenda efetuada no dia 15 de Abril de 2018, foi treinada 2877 vezes. É igualmente importante referir que todas as linhas de dados para treino possuíam a mesma saída pretendida, os tais *targets* estipulados para cada perspetiva. O código desenvolvido para a função *main* pode também ser visto no anexo G.

O algoritmo seguido para o treino da rede neuronal descrito ao longo da secção 4.6 está representado de forma esquemática na Figura 4.5.

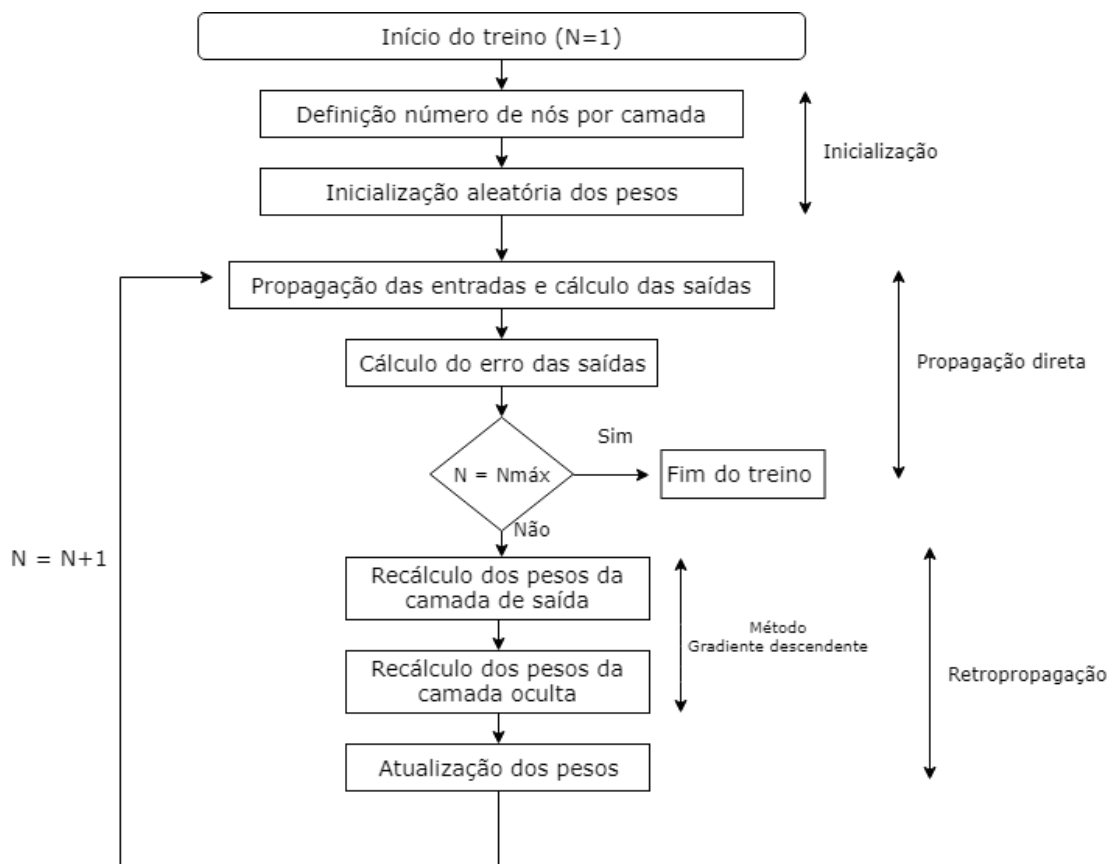


Figura 4.5: Algoritmo do processo de treino

4.6.5 Resultados

Assim que a rede neuronal terminou o seu processo de treino, os resultados obtidos puderam ser analisados. Começando pelo erro total cometido pela rede, era expectável que se fosse aproximando de zero com o decorrer das milhares de iterações, o que se comprovou para duas das perspetivas em análise, o critério de lucro e clientes. A rede foi aprendendo o padrão de entradas e saídas que teriam de ser obtidas e, no final das 2877 iterações, as saídas reais aproximaram-se das saídas desejadas. Tomando como exemplo uma das 2877 linhas de dados, cuja a entrada tinha o valor de 0.1847 para lucro, 0.25 para parceiros e 0.1828 para clientes, as saídas reais obtidas após o treino foram 0.0693, 0.55 e 0.0940, respetivamente. Ora, as saídas pretendidas eram, como se sabe, 0.0693 (\$100 na escala original), 0.7 (70% na escala original) e 0.0940 (5 dias na escala original), o que mostra claramente que as saídas respeitantes a lucro e clientes alcançaram as saídas desejadas, ao contrário do que aconteceu com o critério de parceiros.

No que se refere aos pesos, o seu ajuste iteração a iteração resultou em valores finais que, no fundo, são os valores ideais para que as saídas pretendidas sejam atingidas. Para a camada intermédia foram obtidos os seguintes valores: $w_1 = 4,2768e-15$ para a ligação entre o neurónio 1

e o neurónio 4, $w_2 = 5,1047\text{e-}16$ para a ligação entre o neurónio 2 e o neurónio 4, $w_3 = 3,7948\text{e-}15$ para a ligação entre o neurónio 3 e o neurónio 4 e 0,8764 para a ligação entre o nó extra *bias* e o neurónio 4. Para a camada de saída, os resultados foram os seguintes: $w_4 = 0,0985$ para a conexão entre o neurónio 4 e 5, $w_5 = 1,2309$ para a conexão entre o neurónio 4 e 6 e $w_6 = 0,1338$ para a ligação entre o neurónio 4 e 7. Os valores realmente importantes para a estrutura de otimização desenvolvida até então são os pesos da camada de saída (w_4 , w_5 e w_6) que serão, na secção seguinte, aplicados à hipótese definida no início desta mesma estrutura para validação e verificação do poder da solução ótima encontrada. Os resultados descritos podem ser analisados no anexo H.

4.7 Validação da Hipótese

A última etapa da estrutura, a validação da hipótese, tinha como grande objetivo colocar em prática a solução obtida através do método preditivo, reestruturando a hipótese definida e apresentada na equação 4.14 através da alteração das ponderações iniciais atribuídas a cada perspectiva (33,3%) pelas respetivas ponderações finais que resultaram do processo de treino da rede neuronal. Foram elas: $w_4 = 0,0985$ para lucro, $w_5 = 1,2309$ para parceiros e $w_6 = 0,1338$ para clientes. As métricas associadas a cada perspectiva mantiveram-se, o que significa que a etapa de modelação não sofreu alterações, fazendo a estrutura saltar para a simulação de dados, onde aqui as métricas foram determinadas novamente com a diferença de que cada perspectiva possuiu uma importância relativa distinta.

Os resultados obtidos para cada perspectiva em termos de proximidade com os valores ambicionados, expostos no anexo I, mostram que o lucro médio por mercado variou entre os \$0 para *Africa and India* e os \$6,831 para *The United States and America*, valores abaixo do *target* correspondente de \$9,85, mas com um erro entre os [3,02; 9,85], o que significa uma amplitude de 6,831, bastante inferior aos 19,06 obtidos para um peso de 0,333 e referidos na secção 4.5.3. O mesmo aconteceu com o tempo médio de entrega, com valores a variar entre os 0 e 0,585 dias, abaixo do *target* de 0,669 dias, mas com uma amplitude de intervalo de erro igualmente inferior à registada para a situação inicial, 0,584 contra 1,3048.

No que se refere às vendas médias *online* por estação e por família de produto, a situação é distinta, na medida em que em ambos os casos se registou um aumento da amplitude do intervalo de erro perante a nova solução. Para o caso das vendas *online* por estação, a amplitude obtida foi de 72,486 face aos 19,433 alcançados para a solução inicial, enquanto que no caso das vendas por família de produto a amplitude subiu de 12,23 para 45,618.

À primeira vista aquilo que se pretendia era uma diminuição da amplitude de erro nas três perspetivas, no entanto, não é de estranhar que a perspectiva de parceiros tenha registado um aumento uma vez que, na realidade, o objetivo sempre foi encontrar a melhor relação entre as três visões de negócio, existindo necessariamente um equilíbrio de resultados, ou seja, diminuir a distância para o *target* em lucro e em clientes implicou aumentar essa distância para parceiros.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusão Final

Uma empresa que não atinge as expectativas dos seus clientes é considerada uma empresa dominada pelo fracasso. O consumidor da moda de luxo é um consumidor exigente e não espera nada mais do que um serviço de altíssima qualidade. Uma entrega rápida, uma encomenda bem embalada e um atendimento de primeira linha são requisitos básicos no setor da moda de luxo. Reconhecendo a importância das lojas parceiras, a *Farfetch* tem especial cuidado e atenção para com as mesmas, tentando sempre inovar no sentido de promover uma melhor integração das suas relações e atividades. O foco principal da empresa é expandir os seus negócios com o objetivo de aumentar as suas receitas e, para isso, é inevitável não olhar para clientes e parceiros, os dois principais e grandes motores do modelo de negócio desta plataforma de *e-commerce*.

Um dos principais problemas ou dificuldade com que a empresa se depara, mais concretamente o departamento de produto que lida, entre outras tarefas, com a gestão de uma encomenda, está na escolha do parceiro para atendimento de um pedido já que são vários, por norma, aqueles que o podem fazer espalhados pelo mundo. O presente projeto tinha como principal objetivo construir uma estrutura de modelação, simulação e otimização que permitisse avaliar impactos ao nível de lucro, parceiros e clientes de forma a que se chegasse à melhor relação ou importância relativa entre as três perspetivas, importância essa que seria posta em prática no momento da seleção do melhor fornecedor, tarefa esta executada em *live* no sistema da *Farfetch*.

A revisão de literatura, como em qualquer projeto desta natureza, revelou-se fundamental uma vez que permitiu, em primeiro lugar, compreender o funcionamento mais detalhado de uma cadeia de abastecimento e os problemas que enfrentam consoante o tipo de indústria em que se inserem e, principalmente, que tipo de variáveis são utilizadas na modelação dos seus problemas, assim como as abordagens que são seguidas em termos de simulação e otimização. Destacaram-se os modelos de programação linear, não linear, estocásticos, incertos, metaheurísticas, abordagens *fuzzy* e redes neuronais. Todos revelaram particularidades nas suas aplicações e, dependendo do objetivo que se pretenda, uns podem ser mais eficientes do que outros.

Antes de avançar com a conceção e implementação da estrutura, a situação atual da empresa

no âmbito do projeto necessitava de ser compreendida. Identificar o percurso que uma encomenda faz a partir do momento em que entra nos quadros da *Farfetch*, perceber a metodologia seguida na alocação de *stock points* às encomendas dos clientes e analisar limitações e oportunidades de melhoria nessa estratégia foram os principais aspetos de estudo.

A estrutura foi idealizada tendo em consideração as necessidades da equipa de operações intimamente ligada à análise de dados e consistiu em cinco grandes fases: modelação, simulação, avaliação, otimização e validação. A modelação exigiu obter algum conhecimento acerca da forma como o negócio progride em três frentes: lucro, parceiros e clientes e quais as respetivas variáveis em jogo. Os custos de transporte, algo a minimizar para a empresa e onde são feitas parcerias com a DHL e UPS em consonância com a rota origem-destino, comissões acordadas com as lojas que permitem um ganho para a *Farfetch* com a venda de um produto, assim como a *geomargin* que deve também ser maximizada, são métricas cruciais para as receitas da empresa. Por falar em parceiros, estes devem conseguir vender, devem crescer de uma forma sustentável, porque isso só beneficia a *Farfetch*, interessando-lhe sobretudo a percentagem de vendas que são realizadas a partir do portal, por oposição às vendas *offline*. Mas isto por si só não é suficiente para que esta plataforma consiga melhorar a sua performance dia após dia uma vez que o grande protagonista, o cliente, deve ficar totalmente agradado com o serviço prestado. A começar pelo tempo que a encomenda demora a chegar às suas mãos, algo que foi analisado neste projeto.

A simulação foi realizada em *Atom* e permitiu analisar todas as variáveis anteriormente estudadas com aplicação em encomendas passadas com o intuito de avaliar o padrão de impacto que existiu ao nível de lucro, parceiros e clientes. Impacto esse medido por comparação com valores que seriam os desejados pela empresa. Os resultados revelaram níveis abaixo do pretendido, o que já era de esperar, significando que havia margem para melhorar, daí a existência deste projeto e do sentido da próxima fase da estrutura: a otimização.

O método preditivo, tal como a simulação, foi desenvolvida em *Atom* e a sua implementação foi crucial para atingir o grande objetivo do projeto. A ideia era minimizar a distância existente entre os valores reais devolvidos pela simulação e os valores pretendidos. Para isso, a metodologia seguida foi uma rede neuronal *feedforward* que devolveu como resultado principal as ponderações a atribuir a cada perspetiva analisada que, no plano teórico, iriam permitir uma escolha de parceiro adequada no sentido de aproximar as métricas do plano desejável. A verificação na prática surge na última fase, a validação, onde as métricas foram novamente simuladas, agora sob influência de ponderações distintas. Os resultados mostraram, de facto, que os valores pretendidos para lucro e clientes ficaram mais perto de serem alcançados, contrariamente à perspetiva de parceiros que foi prejudicada.

Os objetivos propostos para a concretização deste projeto foram atingidos e os resultados alcançados foram satisfatórios. Atualmente, em produto, a única simulação que existe é realizada através do método de *A/B testing*, um procedimento poderoso mas que demora bastante tempo e que não permite antecipar impactos na seleção dos fornecedores. Com a implementação da simulação e otimização propostas, este impacto pode ser analisado, como vimos, e o método de *A/B testing* pode ser evitado ou usado apenas como uma confirmação. Assim, torna-se possível

diminuir o chamado *time to market*, um aspeto muito importante na indústria de *e-commerce*.

5.2 Trabalho Futuro

O mercado de *e-commerce* está em constante evolução e a *Farfetch*, como grande empresa da área, deseja evoluir cada vez mais e a todos os níveis. Dada a complexidade do seu modelo de negócio, a estrutura desenvolvida pode e deve ser enriquecida e explorada continuamente nos seguintes aspetos:

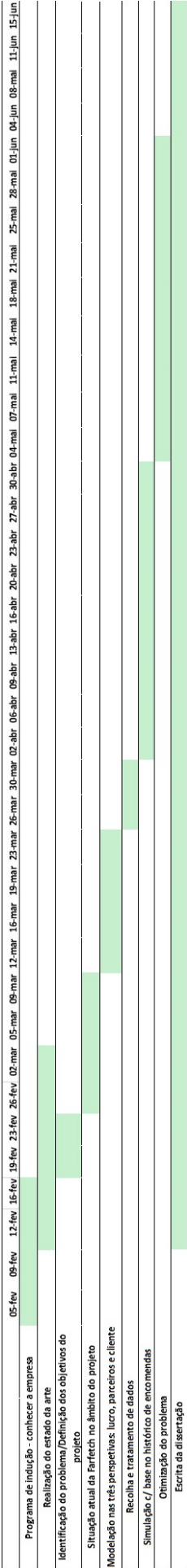
- **Novas perspetivas:** Para além de lucro, clientes e parceiros, poderão ser estudadas e incluídas na hipótese definida novas visões de negócio que se revelem igualmente importantes com a progressão do mercado, com as necessidades da empresa e do meio envolvente mas nunca perdendo o foco do principal objetivo que é a eleição do melhor fornecedor para determinada encomenda.
- **Novas métricas:** Estudar novas métricas adicionais àquelas que foram consideradas e que possam ser incluídas na estrutura de otimização. Com vista a maximizar o lucro por ordem, levar em consideração a profundidade de *stock* dos parceiros, por exemplo. De modo a maximizar o potencial das lojas, estudar a sua taxa de retornos e a pontuação atribuída pelo cliente (NPS) e com o intuito de surpreender ainda mais o cliente, minimizar as taxas adicionais que são pagas aquando da compra de um artigo.
- **Novas ponderações iniciais:** Com o intuito de analisar novos cenários e avaliar os seus impactos na estratégia da *Farfetch*, as ponderações iniciais atribuídas a cada perspetiva podem variar. Isto significa que ao invés de possuírem a mesma importância na decisão de escolha do melhor parceiro, uma poderá ter mais peso em detrimento de outra.
- **Live:** Aplicar a estrutura desenvolvida no sistema da *Farfetch*, mais concretamente, reformular o polinómio atualmente utilizado no serviço de roteamento para seleção de *stock points*, introduzindo perspetivas e métricas abordadas e, mais importante, as ponderações obtidas através do processo de otimização.

Referências

- [1] Min Huang *et al.* «4PL routing optimization under emergency conditions». Em: *Knowledge-Based Systems* (2015). ISSN: 09507051. DOI: [10.1016/j.knosys.2015.06.023](https://doi.org/10.1016/j.knosys.2015.06.023).
- [2] Chen Tung Chen, Ching Torng Lin e Sue Fn Huang. «A fuzzy approach for supplier evaluation and selection in supply chain management». Em: *International Journal of Production Economics* (2006). ISSN: 09255273. DOI: [10.1016/j.ijpe.2005.03.009](https://doi.org/10.1016/j.ijpe.2005.03.009).
- [3] Chin Nung Liao e Hsing Pei Kao. «An integrated fuzzy TOPSIS and MCGP approach to supplier selection in supply chain management». Em: *Expert Systems with Applications* (2011). ISSN: 09574174. DOI: [10.1016/j.eswa.2011.02.031](https://doi.org/10.1016/j.eswa.2011.02.031).
- [4] *Análise Fatorial, uma importante técnica multivariada*. <http://www.abgconsultoria.com.br/blog/analise-fatorial>. Última visita: 3 de Abril de 2018.
- [5] Velu C M e Kashwan K R. «Artificial Neural Network Based Data Mining Technique for Customer Classification for Market Forecasting». Em: (2015).
- [6] Ying Yu *et al.* «E-commerce Logistics in Supply Chain Management: Practice Perspective». Em: *Procedia CIRP*. 2016. DOI: [10.1016/j.procir.2016.08.002](https://doi.org/10.1016/j.procir.2016.08.002).
- [7] *Farfetch: A Empresa do Futuro*. <http://www.openlimits.pt/pt/thinking-ahead-blog/farfetch-empresa-futuro-omnichannel/?all=1>. Última visita: 3 de Abril de 2018.
- [8] *Farfetch: A empresa que funde tecnologia e moda de luxo*. <http://saldopositivo.cgd.pt/empresas/farfetch-empresa-que-funde-tecnologia-e-moda-de-luxo/>. Última visita: 3 de Abril de 2018.
- [9] Kannan Govindan *et al.* «Fuzzy multi-objective approach for optimal selection of suppliers and transportation decisions in an eco-efficient closed loop supply chain network». Em: *Journal of Cleaner Production* (2017). ISSN: 09596526. DOI: [10.1016/j.jclepro.2017.06.180](https://doi.org/10.1016/j.jclepro.2017.06.180).
- [10] Tamás Bányai, Péter Veres e Béla Illés. «Heuristic supply chain optimization of networked maintenance companies». Em: *Procedia Engineering*. 2015. DOI: [10.1016/j.proeng.2015.01.341](https://doi.org/10.1016/j.proeng.2015.01.341).

- [11] *How to Reduce Costs through Supply Chain Network Optimization*. <https://www.linkedin.com/pulse/how-reduce-costs-through-supply-chain-network-stanley-j-oakley>. Última visita: 4 de Abril de 2018.
- [12] Nathalie Silva *et al.* «Improving Supply Chain Visibility With Artificial Neural Networks». Em: *Procedia Manufacturing* (2017). ISSN: 23519789. DOI: [10.1016/j.promfg.2017.07.329](https://doi.org/10.1016/j.promfg.2017.07.329).
- [13] Paulo Bertaglia. «Logística e Gerenciamento da Cadeia de Abastecimento - BERTAGLIA». Em: (2009), pp. 1–35.
- [14] *Make-Or-Buy decision*. <https://www.investopedia.com/terms/m/make-or-buy-decision.asp>. Última visita: 3 de Abril de 2018.
- [15] A. Prakash *et al.* «Network optimization in supply chain: A KBGA approach». Em: *Decision Support Systems* (2012). ISSN: 01679236. DOI: [10.1016/j.dss.2011.10.024](https://doi.org/10.1016/j.dss.2011.10.024).
- [16] Celia Frank *et al.* «NTC Project: S01-PH10 (formerly I01-P10) 1 Forecasting Women's Apparel Sales Using Mathematical Modeling». Em: (2003). URL: <http://faculty.philau.edu/frankc/ntc>.
- [17] *Operadores logísticos e a cadeia de serviços logísticos*. <https://pt.linkedin.com/pulse/operadores-logisticos-e-cadeia-de-servicos-heriton-rodrigues-arantes>. Última visita: 3 de Abril de 2018.
- [18] A Bittante, F Pettersson e H Sax En. «Optimization of a small-scale LNG supply chain». Em: (2018). DOI: [10.1016/j.energy.2018.01.120](https://doi.org/10.1016/j.energy.2018.01.120).
- [19] Ana Isabel Cerqueira De Sousa *et al.* «Otimização de Cadeias de Abastecimento Engenharia e Gestão Industrial». Em: (2014).
- [20] Wan Yu Liu, Chun Cheng Lin e Tzu Lei Yeh. «Supply chain optimization of forest biomass electricity and bioethanol coproduction». Em: *Energy* (2017). ISSN: 03605442. DOI: [10.1016/j.energy.2017.08.018](https://doi.org/10.1016/j.energy.2017.08.018).
- [21] P. C.L. Hui e T. M. Choi. «Using artificial neural networks to improve decision making in apparel supply chain systems». Em: *Information Systems for the Fashion and Apparel Industry*. 2016. ISBN: 9780081005804. DOI: [10.1016/B978-0-08-100571-2.00005-1](https://doi.org/10.1016/B978-0-08-100571-2.00005-1).
- [22] *Você sabe o que é o 4PL?* <http://fgl.com.br/voce-sabe-o-que-e-o-4pl/7>. Última visita: 3 de Abril de 2018.
- [23] *what is 1PL,2PL,3PL and 4PL LOGISTICS? | RASVI*. <http://rasvimk.blogspot.com/2013/11/what-is-1pl2pl3pl-and-4pl-logistics.html>. Última visita: 20 de Fevereiro de 2018.

Anexo A: Diagrama de Gantt



Anexo B: *Queries*

Query Usada para Criar Nova Tabela na Base de Dados

```
IF OBJECT_ID ('BI_ANALYTICS.[FARFETCH\joao.marinho].orders', 'U') IS NOT NULL
    DROP TABLE BI_ANALYTICS.[FARFETCH\joao.marinho].orders;
CREATE TABLE BI_ANALYTICS.[FARFETCH\joao.marinho].orders (
    OrigCountryCode varchar (255) ,
    DestCountryCode varchar (255) ,
    product varchar (255) , DEFAULT ('L') ,
    currency varchar (255) , DEFAULT ('GBP') ,
    [Weight(kg)] FLOAT );
INSERT INTO BI_ANALYTICS.[FARFETCH\joao.marinho].orders
    (OrigCountryCode, DestCountryCode, [Weight(kg)])
VALUES {0[0]} ;
```

Query Usada para Obter os Dados Acerca dos Custos de Transporte Associados à DHL Express

```
select dh1.[Weight(kg)]
    ,dh1.[OrigCountryCode]
    ,dh1.[DestCountryCode]
    ,round(1.1175*PricesDHL.[Values],2) as ExpectedCharge
from [BI_Analysts].[FARFETCH\joao.marinho] orders.dh1
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Zones] zones on
    dh1.origcountrycode = zones.fromcountrycode and
    dh1.destcountrycode = zones.tocountrycode
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Services] [services] on
    [services].sk_service = zones.sk_service
inner join [BI_ANALYTICS].[dbo].[OPS_Del_IR_services_aux] [services_aux] on
    [services_aux].sk_service = zones.sk_service
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Type] [type] on [type].sk_type = zones.sk_type
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_PricesDHL] [pricesDHL] on
```

```

[pricesDHL].zones = zones.zone and
[pricesDHL].sk_service = zones.sk_service and
[pricesDHL].sk_type = zones.sk_type
where dhl1.Currency = 'GBP'
and zones.PriceDate = '2017-01-09'
and [pricesDHL].PriceDate = '2017-01-09'
and [services].sk_service = 7
and carrier_code = 'U'
and dhl1.[Weight(kg)] > [pricesDHL].[Weight_From] and dhl1.[Weight(kg)] <= [pricesDHL].[Weight_to]
order by [Weight(kg)]

```

Exemplo de um Excerto do Resultado Devolvido pela *Query* Anterior

Weight(kg)	OrigCountryCode	DestCountryCode	ExpectedCharge
0,5	GB	HR	10,05
0,5	US	US	3,35
1	ES	HR	13,56
1	IT	AE	13,56
1,3	CY	AT	11,12
1,3	IT	SK	11,12
1,5	RO	VN	15,39
1,5	PL	AU	15,39
1,8	FR	CN	15,91
1,8	PT	CH	18,26
2	PT	VN	16,81
2	IT	AU	16,81
3,5	PL	FR	15,33
3,5	RO	DE	15,33
6	PL	BR	40,69
6	RO	BR	40,69
8,5	SE	GB	25,9
8,5	ES	GB	25,9

Query Usada para Obter os Dados Acerca dos Custos de Transporte Associados à DHL Standard

```
select dhl1.[Weight(kg)]
      ,dhl1.[OrigCountryCode]
      ,dhl1.[DestCountryCode]
      ,round(1.1175*PricesDHL.[Values],2) as ExpectedCharge
from [BI_Analysts].[FARFETCH\joao.marinho] orders.dhl1
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Zones] zones on
      dhl1.origcountrycode = zones.fromcountrycode and
      dhl1.destcountrycode = zones.tocountrycode
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Services] [services] on
      [services].sk_service = zones.sk_service
inner join [BI_ANALYTICS].[dbo].[OPS_Del_IR_services_aux] [services_aux] on
      [services_aux].sk_service = zones.sk_service
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_Type] [type] on [type].sk_type = zones.sk_type
LEFT join [BI_ANALYTICS].[dbo].[OPS_Del_IR_PricesDHL] [pricesDHL] on
      [pricesDHL].zones = zones.zone and
      [pricesDHL].sk_service = zones.sk_service and
      [pricesDHL].sk_type = zones.sk_type
where dhl1.Currency = 'GBP'
      and zones.PriceDate = '2017-01-09'
      and [pricesDHL].PriceDate = '2017-01-09'
      and carrier_code = 'h'
      and dhl1.[Weight(kg)] > [pricesDHL].[Weight_From]
      and dhl1.[Weight(kg)] <= [pricesDHL].[Weight_to]
order by [Weight(kg)]
```

Exemplo de um Excerto do Resultado Devolvido pela *Query* Anterior

Weight(kg)	OrigCountryCode	DestCountryCode	ExpectedCharge
0,5	DE	BE	7,99
0,5	FR	NL	7,99
1,3	SE	GB	12,4
1,3	ES	GB	12,4
1,5	PT	SK	11,32
1,5	RO	AT	11,32
1,8	NL	IT	9,66
1,8	PT	BE	9,66
2	PT	IT	10,62
2	IT	AT	10,62
2,5	LU	AT	9,66
2,5	SE	NL	9,66
3	DE	GB	10,62
3	LU	GB	9,13
5,5	DE	HR	11,32
5,5	FR	SK	11,32
6	PT	NL	9,66
6	DE	ES	9,66
8,5	FR	CH	12,61
8,5	GB	NL	8,57

Query Usada para Criar Nova Tabela na Base de Dados

```
IF OBJECT_ID ('BI_ANALYTICS.[FARFETCH\joao.marinho].ships', 'U') IS NOT NULL
    DROP TABLE BI_ANALYTICS.[FARFETCH\joao.marinho].ships;
CREATE TABLE BI_ANALYTICS.[FARFETCH\joao.marinho].ships (
    zone varchar (255) ,
    [weight] varchar (255) );
INSERT INTO BI_ANALYTICS.[FARFETCH\joao.marinho].ships
    (zone, [weight])
VALUES {0[0]};
```


Query Usada para Obter os Dados Acerca das Zonas e Respetivos Custos Associados à UPS Express

```
select zones
      ,([Values]*0.8) as price
      ,ships.[weight]
from BI_ANALYTICS..[OPS_Del_IR_PricesUPSUS] temp
LEFT join BI_ANALYTICS.[FARFETCH\joao.marinho].ships ships on ships.zone = temp.zones
where ships.[weight] >= weight_from
      and ships.[weight] < Weight_To
```

Exemplo de um Excerto do Resultado Devolvido pela Query Anterior

Zones	price	weight
002	5,856	1.0
002	5,856	2.0
002	8,576	1.0
003	5,856	1.0
003	5,856	2.0
003	8,576	1.0
004	5,856	1.0
004	5,856	2.0
004	8,576	1.0
005	5,856	1.0
005	5,856	2.0
005	8,576	1.0
006	5,856	2.0
006	8,576	1.0
006	8,576	2.0
007	5,856	1.0
007	5,856	2.0
008	5,856	1.0
008	5,856	2.0

Query Usada para Obter os Dados Acerca das Zonas e Respetivos Custos Associados à UPS Standard

```
select zones
      ,([Values]*0.8) as price
      ,ships.[weight]
from BI_ANALYTICS..[OPS_Del_IR_PricesUPSUS] temp
LEFT join BI_ANALYTICS.[FARFETCH\joao.marinho].ships ships on ships.zone = temp.zones
where ships.[weight] >= weight_from
      and ships.[weight] < Weight_To
```

Exemplo de um Excerto do Resultado Devolvido pela Query Anterior

Zones	price	weight
002	5,856	1.0
002	5,856	2.0
002	8,576	1.0
003	5,856	2.0
003	8,576	1.0
003	8,576	2.0
004	5,856	2.0
004	8,576	1.0
004	8,576	2.0
005	5,856	2.0
005	8,576	1.0
005	8,576	2.0
006	5,856	1.0
006	5,856	2.0
006	8,576	1.0
007	8,576	1.0
007	8,576	2.0
008	5,856	1.0
008	5,856	2.0

Query Usada para Obter os Dados Necessários à Realização da Simulação

```
select GLB.OrderCode as PortalOrderId,
       forders.sk_product as Produto,
       X.IdStockPoint as StockPoint,
       X.IdStore as boutique,
       X.Country as StockPointCountry,
       SC.countrycode as StockPointCountryCode,
       GLB.Country as UserCountry,
       GC.countrycode as UserCountryCode,
       market.ContinentFF as Continent,
       EDD.Carrier as Carrier,
       EDD.ServiceCountry as Service,
       GLB.Zip as userzipcode,
       X.zipcode as stockpointzipcode,
       (EDD.Step6Raw + EDD.SOSRaw) as leadtime,
       X.FarfetchnProfitPerc as comissao,
       forders.RetailPrice_USD as RRP,
       forders.GeoMarginComission as Geomargin,
       dhl.[Weight(kg)] as Peso
from BI_Sync..glborders GLB
join (select *
      from [FFLIVE\BI Analysts].OPS_Del_KPI_Time EDD
      where EDD.Step6Raw + EDD.SOSRaw is not null) EDD on GLB.OrderID = EDD.OrderID
join BI_Sync..globalcountries GC on GLB.countryid = GC.countryid
join (select distinct paisid, continentFF
      from BIDW.BI_DW.dbo.DimGeography market) market on GLB.countryid = market.paisid
join (select *
      from BIDW.BI_DW.dbo.FactOrdersLines forders
      where forders.GeoMarginComission is not null) forders on GLB.glborderid = forders.orderlineid
cross apply (select AD.*, SP.*, store.farfetchnprofitperc
            from BI_SYNC..StockPoint SP
            inner join BI_SYNC..AddressInfo AI on SP.IdShippingAddress = AI.IdAddressInfo
            inner join BI_SYNC..Address AD on AI.IdAddress = AD.IdAddress
            inner join (select *
                      from BIDW.BI_DW.dbo.DimStore store
                      where store.FarfetchnProfitperc is not null) store on SP.StockPointKey = store.SK_store
            where SP.stockpointkey = GLB.siteid) X
join BI_Sync..globalcountries SC on X.countryid = SC.countryid
```

```

join BIDW.BI_DW.dbo.FactDHLBilling dhl on orders.AWB = dhl.ShipmentNumber
    and (dhl.XC1Code like '%FF%' or dhl.XC1Code like '%FB%')
    and abs (DATEDIFF (dd, Convert (datetime, LEFT (dhl.ShipmentDate, 8)), orders.SendDate
)) <= 30
where GLB.DataCriado >= '2018-04-15'
    and GLB.DataCriado < '2018-04-16'
    and EstimatedDeliveryDate is not null
    and EDD.siteid = GLB.SiteID
order by GLB.DataCriado ASC

```

Exemplo de um Excerto do Resultado Devolvido pela Query Anterior

PortalOrderId	Product	StockPoint	boutique	StockPointCountry	StockPointCountryCode
VYVX5B	3347810	1907	1654	United Kingdom	GB
KSKP67	4190651	769	729	Italy	IT
7M72MZ	3447013	1323	771	United Kingdom	GB
U5U8PV	4631197	500	465	Italy	IT
F8FKUM	3416498	1323	771	United Kingdom	GB
EHESJG	3269814	1151	1100	Italy	IT
G9GV66	4326301	425	393	Germany	DE
3T32JS	4491505	1059	1012	United States	US
XGXN2Y	4632012	1158	1107	Italy	IT
B4BWFX	4547048	495	460	Denmark	DK
B4BWFX	4294076	567	533	Germany	DE
WRWJKU	3334154	2076	1778	Italy	IT
B4BWHX	4737603	923	878	Japan	JP
B4BWHX	4644950	1907	1654	United Kingdom	GB
EHE3JG	3370173	581	547	Germany	DE
PFP543	4710025	550	516	Spain	ES
G9GVB6	4500817	535	501	Germany	DE
G9GVB6	4633302	2148	496	Spain	ES
LXLCAP	4390612	1323	771	United Kingdom	GB
LXLCAP	4449599	1000	953	United States	US
7M74KZ	3627345	1934	1684	United States	US
5K5DGA	3347309	1569	1406	Italy	IT

UserCountry	UserCountryCode	Continent	Carrier
Canada	CA	The United States & Canada	DHL Express
Korea, Republic of	KR	Asia Pacific	DHL Express
UAE	AE	Middle East	DHL Express
United States	US	The United States & Canada	DHL Express
Brazil	BR	Latin America and the Caribbean	DHL Express Brazil
United States	US	The United States & Canada	DHL Express
Australia	AU	Asia Pacific	DHL Express
United States	US	The United States & Canada	UPS Ground
Japan	JP	Asia Pacific	DHL Express
Russian Federation	RU	Europe	DHL Express
Russian Federation	RU	Europe	DHL Express
Japan	JP	Asia Pacific	DHL Express
United States	US	The United States & Canada	DHL Express
United States	US	The United States & Canada	DHL Express
Brazil	BR	Latin America and the Caribbean	DHL Express Brazil
United States	US	The United States & Canada	DHL Express
Korea, Republic Of	KR	Asia Pacific	DHL Express
Korea, Republic Of	KR	Asia Pacific	DHL Express
United States	US	The United States & Canada	DHL Express
United States	US	The United States & Canada	UPS Ground
United States	US	The United States & Canada	UPS Ground
Japan	JP	Asia Pacific	DHL Express

Service	userzipcode	stockpointzipcode	leadtime	comission	RRP	Geomargin	weight
E	11r2b3	EN3 7SN	4,9	21	163.9410	0.00000	0,9
E	6549	21052	3,4	25	417.6340	0.00000	3,6
E	900090	EN37SN	3,3	25	928.6520	0.00000	4,9
E	2453	81030	3,8	24,4	181.5800	0.00000	0,9
E	11370-130	EN37SN	4,6	25	307.8830	0.00000	4
E	30342	91100	4,8	26,8	346.6920	14.33700	3,5
E	2092	88131	5	18	1760.2420	356.29597	3,5
S	11754	7310	9,8	22	425.6980	0.00007	2
E	153-0063	25020	3,5	0	138.8080	0.00000	1
E	119034	2100	3,4	26,8	561.4620	0.00000	2
E	119034	73614	3,3	0	179.0690	0.00000	2
E	491-0911	96100	7,4	26,8	282.0440	34.50914	2,5
E	76120	107-0062	8,9	25	193.5390	0.00000	2
E	76120	EN3 7SN	8,9	21	204.5600	0.00000	2
E	05608-020	79098	3,6	25	621.2890	6.05162	7
E	10001	32003	2,8	22	219.1710	0.00000	3,5
E	7236	3180	3,1	22	91.2750	0.00003	2
E	7236	59555	3,1	26,18	1418.1760	0.00000	1
E	77057	EN37SN	3,1	25	156.5510	0.00000	5,5
S	77057	10001	6,1	25	85.4030	0.00000	3,5
S	48105	7604	12,9	26,04	122.1100	0.00000	3
E	158-0096	91100	6	22	356.0720	34.87224	3,5

Query Usada para Obter os Dados Acerca de *Sell Through* e Peso das Vendas *Online*

```
select fst.sk_product as Product,
       fst.year,
       fst.WeekNumber,
       fst.QtyNetSold_All,
       fst.MaxUsefullStock,
       fst.StockEndOfWeek,
       fst.QtySalesOffline_All,
       fst.Season,
       dpc.Category,
from   dbo.FactStockSellThru fst
inner join FactOrdersLines fact on fact.sk_product = fst.sk_product
inner join dimproduct p on fst.sk_product = p.sk_product
inner join FactProductCategory pc on pc.sk_product = p.sk_product
inner join DimProductCategory dpc on dpc.sk_category = pc.sk_CategoryAnalysis
       and dpc.IsMainCategoryAnalysis = 'Yes'
where  sk_SellThruDate = '20180415'
```

Exemplo de um Excerto do Resultado Devolvido pela Query Anterior

Product	Year	WeekNumber	QtyNetSold_All	MaxUsefullStock
4723521	2018	15	1	10
1726570	2018	15	0	0
2765873	2018	15	0	11
2615093	2018	15	1	3
3362314	2018	15	0	5
4255751	2018	15	12	20
4119334	2018	15	6	9
4644988	2018	15	1	7
4226443	2018	15	0	3
4724819	2018	15	0	2
2576071	2018	15	1	3
3156932	2018	15	0	2
3498182	2018	15	0	6

StockEndOfWeek	QtySalesOffline_All	Season	Category	Family
6	3	SS18	Loafers	Shoes
0	0	SS16	Wallets & Purses	Accessories
0	11	SS17	Sweaters & Knitwear	Clothing
0	2	SS17	Pumps	Shoes
0	5	AW17	Backpacks	Bags
0	8	AW17	Trainers	Shoes
0	3	AW17	Tops	Boys Clothing
5	1	SS18	T-Shirts & Vests	Clothing
0	3	AW17	Sweaters & Knitwear	Clothing
2	0	SS18	Jackets	Clothing
0	2	SS16	T-Shirts & Vests	Clothing
0	2	SS17	Dresses	Clothing
0	6	AW17	Denim	Clothing

Anexo C: Dados Externos

Excerto de Zonas para a UPS *Express*

tozip	fromzip	zone
010	100	202
010	331	206
012	331	206
014	076	203
016	303	205
016	333	206
017	212	204
017	900	208
018	946	208
019	070	203
019	073	203
020	303	205
020	941	208
021	021	202
021	070	203
029	980	208
031	076	203
032	100	203
037	076	203

Excerto de Zonas para a UPS *Standard*

tozip	fromzip	zone
770	949	007
770	972	008
770	980	008
770	981	008
772	100	006
772	902	006
773	068	006
773	070	006
773	073	006
773	076	006
813	070	007
813	100	007
813	110	007
813	900	005
814	076	007
986	980	002
988	100	008
991	076	008
991	078	008

Anexo D: Código da Simulação

Ligação à Base de Dados

```
1 import os
2 import pyodbc
3 import subprocess
4 import sys
5
6 os.chdir('c:\\Users\\diogo.fernandes\\Documents\\sql')
7
8 def read_query_file(path, *var):
9     path= "sql/" + path
10    file_object = open(path, 'r')
11    sqlFile = str(file_object.read())
12    sqlFile=sqlFile.format(var,)
13    #print(sqlFile)
14    connection = pyodbc.connect("Driver={SQL Server Native Client 11.0};"
15                                "Server=bidw.farffetch.net;"
16                                "Database=bi_dw;"
17                                "Trusted_Connection=yes;")
18    cursor = connection.cursor()
19    cursor.execute(sqlFile)
20    return cursor
21
22 def read_query_file_com(path, *var):
23     path = "sql/" + path
24
25     file_object = open(path, 'r')
26     sqlFile = str(file_object.read())
27     sqlFile = sqlFile.format(var,)
28
29
30     connection = pyodbc.connect("Driver={SQL Server Native Client 11.0};"
31                                "Server=bidw.farffetch.net;"
32                                "Trusted_Connection=yes;")
33     cursor = connection.cursor()
34
35     cursor.execute(sqlFile)
36     cursor.commit()
37     return cursor
```

Consulta de Dados para Custos de Transporte DHL

```
cursor = read_query_file_com("DHL_ship_Insert.sql")
for i in range(1, iter):
    cursor = read_query_file_com("DHL_ship_Insert2.sql",)

cursor = read_query_file("DHL_ship_Express.sql")
rows = cursor.fetchall()
names = [column[0] for column in cursor.description]

dhl_express = pd.DataFrame(np.array(rows), columns=names)
return dhl_express

cursor = read_query_file("DHL_ship_Standard.sql")
rows = cursor.fetchall()
names = [column[0] for column in cursor.description]

dhl_standard = pd.DataFrame(np.array(rows), columns=names)
return dhl_standard
```

Consulta de Dados para Custos de Transporte UPS

```
#zones
UPS_zones_standard=pd.read_excel('data/UPS_standard_zones.xlsx',converters={'tozip':str,'fromzip':str,'zone':str})
UPS_zones_express=pd.read_excel('data/UPS_express_zones.xlsx',converters={'tozip':str,'fromzip':str,'zone':str})

#express
temp_values=tuple(ups_express_prices.loc[:,["zone","Weights"]].values.tolist())
temp_values=clean_list(temp_values)
temp_values

cursor=read_query_file_com("UPS_ship_Insert.sql",temp_values)
cursor=read_query_file("UPS_ship_Express.sql")

names= [column[0] for column in cursor.description]
rows=cursor.fetchall()
ups_express_prices = pd.DataFrame(np.array(rows), columns=names)

#standard
temp_values=tuple(ups_standard_prices.loc[:,["zone","Weights"]].values.tolist())
temp_values=clean_list(temp_values)
temp_values

cursor=read_query_file_com("UPS_ship_Insert.sql",temp_values)
cursor=read_query_file("UPS_ship_Standard.sql")

names= [column[0] for column in cursor.description]
rows=cursor.fetchall()
ups_standard_prices = pd.DataFrame(np.array(rows), columns=names)
```

Lógica da Simulação

```
import os
os.chdir('c:\\Users\\diogo.fernandes\\Documents\\sql')

import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile

ups_standard_prices = pd.read_excel('ups_standard_prices.xlsx')
ups_express_prices = pd.read_excel('ups_express_prices.xlsx')
UPS_standard_zones = pd.read_excel('UPS_standard_zones.xlsx')
UPS_express_zones = pd.read_excel('UPS_express_zones.xlsx')
sellthru = pd.read_excel('sellthru2.xlsx')

import historicalorders
import DHLshippingCosts
import numpy as np
import pyodbc
import numpy
import math
import random
import csv

from random import seed
from random import random
from math import exp
import numpy as np
import numpy
import string

try:
    import pyodbc
except ImportError:
    import odbc as pyodbc

cursor = historicalorders.read_query_file("myquery.sql")
cursor1 = DHLshippingCosts.read_query_file("DHL_ship_Express.sql")
cursor2 = DHLshippingCosts.read_query_file("DHL_ship_Standard.sql")

rows = cursor.fetchall()
names = [column[0] for column in cursor.description]
myorders = pd.DataFrame(np.array(rows), columns=names)

rows1 = cursor1.fetchall()
names1 = [column[0] for column in cursor1.description]
orders = pd.DataFrame(np.array(rows1), columns=names1)

rows2 = cursor2.fetchall()
names2 = [column[0] for column in cursor2.description]
orders1 = pd.DataFrame(np.array(rows2), columns=names2)
```

```

profit = 0.33
customers = 0.33
partners = 0.33

profit_target = 100
customers_target = 5
partners_target = 70

i = 0
j = 0
w = 0

for x in range(0, len(rows)):

    if((myorders.loc[i]['Service'] == 'E') and (myorders.loc[i]['Carrier'] == 'DHL Express' or myorders.loc[i]['Carrier'] == 'DHL Express China' or myorders.loc[i]['Carrier'] == 'DHL Express Brazil')):

        for x in range(0, len(rows1)):
            #DHL
            if((myorders.loc[i]['StockPointCountryCode'] == orders.loc[j]['OrigCountryCode'] and (myorders.loc[i]['UserCountryCode'] == orders.loc[j]['DestCountryCode'] and ((str(myorders.loc[i]['weight'])) == orders.loc[j]['weight'])) == orders.loc[j]['weight(kg)'))):

                results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
                results.loc[i]['Product'] = myorders.loc[i]['Product']
                results.loc[i]['Continent'] = myorders.loc[i]['Continent']
                results.loc[i]['Boutique'] = myorders.loc[i]['Boutique']
                results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
                results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 + float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['Geomargin']))) - float(orders.loc[j]['ExpectedCharge']))
                results.loc[i]['Customers'] = (myorders.loc[i]['leadtime'])

                ST = (sellthru['QtyMetsSold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullStock'][w]
                PesoVendasOnline = (sellthru['QtyMetsSold_All'][w])/((sellthru['QtyMetsSold_All'][w] + sellthru['QtySalesOffline_All'][w]))

                results.loc[i]['Partners'] = (PesoVendasOnline*ST)*100

                results.loc[i]['Season'] = sellthru['Season'][w]
                results.loc[i]['Category'] = sellthru['Category'][w]
                results.loc[i]['Family'] = sellthru['Family'][w]

                targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
                targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
                targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']

            break

        else:
            j = j+1

```

```

elif(myorders.loc[i]['Service'] == 'S') and (myorders.loc[i]['Carrier'] == 'DHL Ground' or myorders.loc[i]['Carrier'] == 'DHL Economy')):
    for x in range(0, len(rows2)):
        if ((myorders.loc[i]['StockPointCountryCode'] == orders1.loc[j]['OrigCountryCode']) and (myorders.loc[i]['UserCountryCode'] == orders1.loc[j]['DestCountryCode']) and ((str(myorders.loc[i]['weight']) == orders1.loc[j]['weight(kg)')))):
            results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
            results.loc[i]['Product'] = myorders.loc[i]['Product']
            results.loc[i]['Continent'] = myorders.loc[i]['Continent']
            results.loc[i]['Boutique'] = myorders.loc[i]['Boutique']
            results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
            results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 * float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['ExpectedCharge'])))
            results.loc[i]['Customers'] = (myorders.loc[i]['Leadtime'])

            ST = (sellthru['QtyNetSold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullStock'][w]
            PesoVendasOnline = (sellthru['QtyNetSold_All'][w])/(sellthru['QtyNetSold_All'][w] + sellthru['QtySalesOffline_All'][w])

            results.loc[i]['Partners'] = (PesoVendasOnline*ST)**100

            results.loc[i]['Season'] = sellthru['Season'][w]
            results.loc[i]['Category'] = sellthru['Category'][w]
            results.loc[i]['Family'] = sellthru['Family'][w]

            targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
            targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
            targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']
        break
    else:
        j = j+1

```

```

elif((myorders.loc[i]['Service'] == 'E') and (myorders.loc[i]['Carrier'] == 'UPS Second Day Air') and (myorders.loc[i]['StockPointCountryCode'] == 'US') and (myorders.loc[i]['UserCountryCode'] == 'US')):
    for x in range(0, len(UPS_express_zones)):
        if((myorders.loc[i]['stockpointzipcode'][:3] == (str(UPS_express_zones['fromzip'][j]))) and (myorders.loc[i]['userzipcode'][:3] == (str(UPS_express_zones['tozip'][j])))):
            zone = UPS_express_zones['zone'][j]
            k = 0
            for x in range(0, len(ups_express_prices)):
                if(zone == ups_express_prices['Zones'][k]):
                    price = ups_express_prices['price'][k]
                else:
                    k = k+1
            results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
            results.loc[i]['Product'] = myorders.loc[i]['Product']
            results.loc[i]['Continent'] = myorders.loc[i]['Continent']
            results.loc[i]['Boutique'] = myorders.loc[i]['boutique']
            results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
            results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 * float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['Geomargin']))) - price)
            results.loc[i]['Customers'] = (myorders.loc[i]['leadtime'])

ST = (sellthru['QtyMetSold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullstock'][w]
PesoVendasOnline = (sellthru['QtyMetSold_All'][w])/((sellthru['QtyMetSold_All'][w] + sellthru['QtySalesOffline_All'][w]))

results.loc[i]['Partners'] = (PesoVendasOnline*ST)*100

results.loc[i]['Season'] = sellthru['Season'][w]
results.loc[i]['Category'] = sellthru['Category'][w]
results.loc[i]['Family'] = sellthru['Family'][w]

targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']

break
else:
    j = j+1

```



```

elif((myorders.loc[i]['Service'] == 'S') and (myorders.loc[i]['Carrier'] == 'UPS Standard' or myorders.loc[i]['Carrier'] == 'UPS Ground') and (myorders.loc[i]['StockPointCountryCode'] == 'US')
and (myorders.loc[i]['UserCountryCode'] == 'US')):

    for x in range(0, len(ups_standard_zones)):

        if ((myorders.loc[i]['stockpointzipcode'][:3] == (str(ups_standard_zones['fromzip'][j]))) and (myorders.loc[i]['userzipcode'][:3] == (str(ups_standard_zones['tozip'][j])))):

            zone = ups_standard_zones['zone'][j]
            k = 0
            for x in range(0, len(ups_standard_prices)):

                if (zone == ups_standard_prices['Zones'][k]):
                    price = ups_standard_prices['price'][k]
                else:
                    k = k+1

            results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
            results.loc[i]['Product'] = myorders.loc[i]['Product']
            results.loc[i]['Continent'] = myorders.loc[i]['Continent']
            results.loc[i]['Boutique'] = myorders.loc[i]['boutique']
            results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
            results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 * float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['Geomangin'])) - price)
            results.loc[i]['Customers'] = (myorders.loc[i]['leadtime'])

            ST = (sellthru['QtyNetSold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullStock'][w]
            PesoVendasOnline = (sellthru['QtyNetSold_All'][w])/(sellthru['QtyNetSold_All'][w] + sellthru['QtySalesOffline_All'][w])

            results.loc[i]['Partners'] = (PesoVendasOnline*ST)*100

            results.loc[i]['Season'] = sellthru['Season'][w]
            results.loc[i]['Category'] = sellthru['Category'][w]
            results.loc[i]['Family'] = sellthru['Family'][w]

            targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
            targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
            targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']

            break

        else:
            j = j+1

```

```

elif(myorders.loc[i]['Service'] == 'E'):
    for x in range(0, len(rows1)):
        if((myorders.loc[i]['StockPointCountryCode'] == orders.loc[i]['OrigCountryCode']) and (myorders.loc[i]['UserCountryCode'] == orders.loc[i]['DestCountryCode']) and ((str(myorders.loc[i]['weight']) == orders.loc[i]['Weight(kg)'])))
            results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
            results.loc[i]['Product'] = myorders.loc[i]['Product']
            results.loc[i]['Continent'] = myorders.loc[i]['Continent']
            results.loc[i]['Boutique'] = myorders.loc[i]['Boutique']
            results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
            results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 * float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['Geomargin']))) - float(orders.loc[j]['ExpectedCharge']))
            results.loc[i]['Customers'] = (myorders.loc[i]['Leadtime'])

            ST = (sellthru['QtyWetsold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullstock'][w]
            PesoVendasOnline = (sellthru['QtyWetsold_All'][w])/(sellthru['QtyWetsold_All'][w] + sellthru['QtySalesOffline_All'][w])

            results.loc[i]['Partners'] = (PesoVendasOnline*ST)*100

            results.loc[i]['Season'] = sellthru['Season'][w]
            results.loc[i]['Category'] = sellthru['Category'][w]
            results.loc[i]['Family'] = sellthru['Family'][w]

            targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
            targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
            targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']

            break
        else:
            j = j+1

```

```

elif(myorders.loc[i]['Service'] == 'S'):
    for x in range(0, len(rows2)):
        if ((myorders.loc[i]['StockPointCountryCode'] == orders1.loc[j]['OrigCountryCode'] and (myorders.loc[i]['UserCountryCode'] == orders1.loc[j]['DestCountryCode'] and ((str(myorders.loc[i]['weight']) == orders1.loc[j]['weight(kg)'])))

            results.loc[i]['Order'] = myorders.loc[i]['PortalOrderId']
            results.loc[i]['Product'] = myorders.loc[i]['Product']
            results.loc[i]['Continent'] = myorders.loc[i]['Continent']
            results.loc[i]['Boutique'] = myorders.loc[i]['boutique']
            results.loc[i]['Stock point'] = myorders.loc[i]['StockPoint']
            results.loc[i]['Profit'] = (((myorders.loc[i]['commission']/100 * float(myorders.loc[i]['RRP'])) + float(myorders.loc[i]['geomargin'])) - float(orders1.loc[j]['ExpectedCharge']))
            results.loc[i]['Customers'] = (myorders.loc[i]['leadtime'])

            ST = (sellthru['QtyMetSold_All'][w] + sellthru['QtySalesOffline_All'][w])/sellthru['MaxUsefullStock'][w]
            PesoVendasOnline = (sellthru['QtyMetSold_All'][w])/((sellthru['QtyMetSold_All'][w] + sellthru['QtySalesOffline_All'][w]))

            results.loc[i]['Partners'] = (PesoVendasOnline*ST)*100

            results.loc[i]['Season'] = sellthru['Season'][w]
            results.loc[i]['Category'] = sellthru['Category'][w]
            results.loc[i]['Family'] = sellthru['Family'][w]

            targets.loc[i]['profit'] = profit_target - results.loc[i]['Profit']
            targets.loc[i]['customers'] = customers_target - results.loc[i]['Customers']
            targets.loc[i]['partners'] = partners_target - results.loc[i]['Partners']

        break
    else:
        j = j+1

j = 0
w = w+1
i = i+1

```

Normalização dos Dados

```
df = results.dropna()
df.to_excel('resultados1.xlsx', sheet_name='results', index=False)

profit_max = df['Profit'].max()
profit_min = df['Profit'].min()
print(profit_max)
print(profit_min)
partners_max = df['Partners'].max()
partners_min = df['Partners'].min()
print(partners_max)
print(partners_min)
customers_max = df['Customers'].max()
customers_min = df['Customers'].min()
print(customers_max)
print(customers_min)

if (profit_max < profit_target):
    targ_profit_norm = 1
else:
    targ_profit_norm = (profit_target - profit_min)/(profit_max - profit_min)

if (partners_max < partners_target):
    targ_partners_norm = 1
else:
    targ_partners_norm = (partners_target - partners_min)/(partners_max - partners_min)

if (customers_max < customers_target):
    targ_customers_norm = 1
else:
    targ_customers_norm = (customers_target - customers_min)/(customers_max - customers_min)

df['Profit'] = (df['Profit'] - profit_min)/(profit_max-profit_min)
df['Partners'] = (df['Partners'] - partners_min)/(partners_max-partners_min)
df['Customers'] = (df['Customers'] - customers_min)/(customers_max-customers_min)

df_final = df.reset_index()
df_final.to_excel('resultados3.xlsx', sheet_name='results', index=False)
```

Anexo E: Resultados da Simulação

Excerto de Resultados na Escala Original

Order	Product	Continent	Boutique	Stock point	Season
VYVX5B	3347810	The United States & Canada	1654	1907	AW17
KSKP67	4190651	Asia Pacific	729	769	SS18
7M72MZ	3447013	Middle East	771	1323	VINTAGE
U5U8PV	4631197	The United States & Canada	465	500	AW16
F8FKUM	3416498	Latin America and the Caribbean	771	1323	AW17
EHESJG	3269814	The United States & Canada	1100	1151	AW17
G9GV66	4326301	Asia Pacific	393	425	AW17
XGXN2Y	4632012	Asia Pacific	1107	1158	AW17
B4BWFx	4547048	Europe	460	495	SS17
WRWJKU	3334154	Asia Pacific	1778	2076	AW17
B4BWHX	4644950	The United States & Canada	1654	1907	SS16
EHE3JG	3370173	Latin America and the Caribbean	547	581	SS18
PFP543	4710025	The United States & Canada	516	550	AW17
LXLCAP	4449599	The United States & Canada	953	1000	SS18
LXLCAP	4390612	The United States & Canada	771	1323	SS15
5K5DGA	3347309	Asia Pacific	1406	1569	CARRYOVER
5K5DGA	4275947	Asia Pacific	519	553	AW17
5K5DGA	4230060	Asia Pacific	1100	1272	VINTAGE
VYVX7B	4338908	The United States & Canada	667	702	SS17
TATE3H	3449999	Latin America and the Caribbean	771	1323	SS18
F8FKCM	4408282	The United States & Canada	1120	1995	AW17

Category	Family	Profit	Customers	Partners
Sweaters & Knitwear	Clothing	7,2691113	1,617	0
Shoulder Bags	Bags	26,841705	1,122	26,4
Vintage Jewellery	Vintage & Archive	67,94139	1,089	33
Trainers	Shoes	11,2251216	1,254	5,5
Hats	Girls Accessories	18,6749475	1,518	8,25
Jackets	Clothing	30,04335048	1,584	16,5
Trainers	Shoes	215,6779449	1,65	11,64705882
Boots	Shoes	-4,5144	1,155	18
Denim	Clothing	44,10839928	1,122	10,15384615
Trainers	Shoes	30,26988756	2,442	3,666666667
Tops	Clothing	9,529608	2,937	22
Shirts	Clothing	38,3670771	1,188	5,076923077
Knitwear	Clothing	10,5625146	0,924	33
Tops	Clothing	4,2156675	2,013	22
Dresses	Clothing	6,5332575	1,023	0
Skirts	Clothing	9,419322	0,99	0
Tops	Clothing	6,44589	1,056	16,5
Vintage Jewellery	Vintage & Archive	35,3364792	1,98	16,5
Skirts	Clothing	6,22979775	1,98	11
Coats	Clothing	-1,2577125	1,155	16,5
T-Shirts & Vests	Clothing	12,643257	1,947	19,8

Excerto de Resultados na Escala Normalizada

Order	Product	Continent	Boutique	Stock point	Season
VYVX5B	3347810	The United States & Canada	1654	1907	AW17
KSKP67	4190651	Asia Pacific	729	769	SS18
7M72MZ	3447013	Middle East	771	1323	VINTAGE
U5U8PV	4631197	The United States & Canada	465	500	AW16
F8FKUM	3416498	Latin America and the Caribbean	771	1323	AW17
EHESJG	3269814	The United States & Canada	1100	1151	AW17
G9GV66	4326301	Asia Pacific	393	425	AW17
XGXN2Y	4632012	Asia Pacific	1107	1158	AW17
B4BWFX	4547048	Europe	460	495	SS17
WRWJKU	3334154	Asia Pacific	1778	2076	AW17
B4BWHX	4644950	The United States & Canada	1654	1907	SS16
EHE3JG	3370173	Latin America and the Caribbean	547	581	SS18
PFP543	4710025	The United States & Canada	516	550	AW17
LXLCAP	4449599	The United States & Canada	953	1000	SS18
LXLCAP	4390612	The United States & Canada	771	1323	SS15
5K5DGA	3347309	Asia Pacific	1406	1569	CARRYOVER
5K5DGA	4275947	Asia Pacific	519	553	AW17
5K5DGA	4230060	Asia Pacific	1100	1272	VINTAGE
VYVX7B	4338908	The United States & Canada	667	702	SS17
TATE3H	3449999	Latin America and the Caribbean	771	1323	SS18
F8FKCM	4408282	The United States & Canada	1120	1995	AW17

Category	Family	Profit	Customers	Partners
Sweaters & Knitwear	Clothing	0,030341303	0,091383812	0
Shoulder Bags	Bags	0,059969893	0,052219321	0,8
Vintage Jewellery	Vintage & Archive	0,12218575	0,049608355	1
Trainers	Shoes	0,03632983	0,062663185	0,166666667
Hats	Girls Accessories	0,047607223	0,083550914	0,25
Jackets	Clothing	0,064816478	0,088772846	0,5
Trainers	Shoes	0,345826305	0,093994778	0,352941176
Boots	Shoes	0,012503667	0,054830287	0,545454545
Denim	Clothing	0,086107859	0,052219321	0,307692308
Trainers	Shoes	0,065159405	0,156657963	0,111111111
Tops	Clothing	0,033763197	0,195822454	0,666666667
Shirts	Clothing	0,077416764	0,057441253	0,153846154
Knitwear	Clothing	0,035326789	0,036553525	1
Tops	Clothing	0,025719063	0,122715405	0,666666667
Dresses	Clothing	0,029227383	0,044386423	0
Skirts	Clothing	0,033596248	0,041775457	0
Tops	Clothing	0,029095128	0,046997389	0,5
Vintage Jewellery	Vintage & Archive	0,072829107	0,120104439	0,5
Skirts	Clothing	0,028768012	0,120104439	0,333333333
Coats	Clothing	0,017433573	0,054830287	0,5
T-Shirts & Vests	Clothing	0,038476574	0,117493473	0,6

Anexo F: Estatísticas

Gráfico Circular sobre a Percentagem de Encomendas por Mercado para um Dia

Number of orders	Continent	Number of orders	Percentage
2877	Africa and India	0	0
	Asia Pacific	877	30,48314216
	Europe	1171	40,70212026
	Middle East	106	3,684393465
	Latin American and the Caribbean	29	1,007994439
	The United States & Canada	694	24,1223497

■ Africa and India
 ■ Asia Pacific
 ■ Europe
 ■ Middle East
 ■ Latin America and the Caribbean
 ■ The United States & Canada

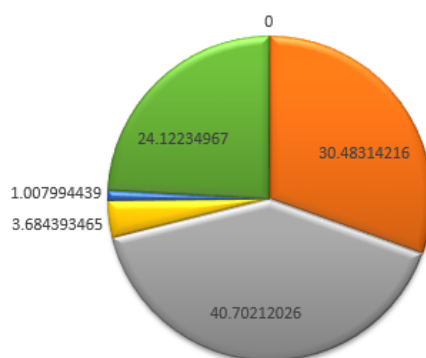


Gráfico de Barras sobre o Lucro Médio por Mercado para um Dia

Number of orders	Continent	Average profit	Target
2877	Africa and India	0	33
	Asia Pacific	19,05709435	33
	Europe	19,00308268	33
	Middle East	18,99793838	33
	Latin American and the Caribbean	18,94107918	33
	The United States & Canada	19,06128995	33

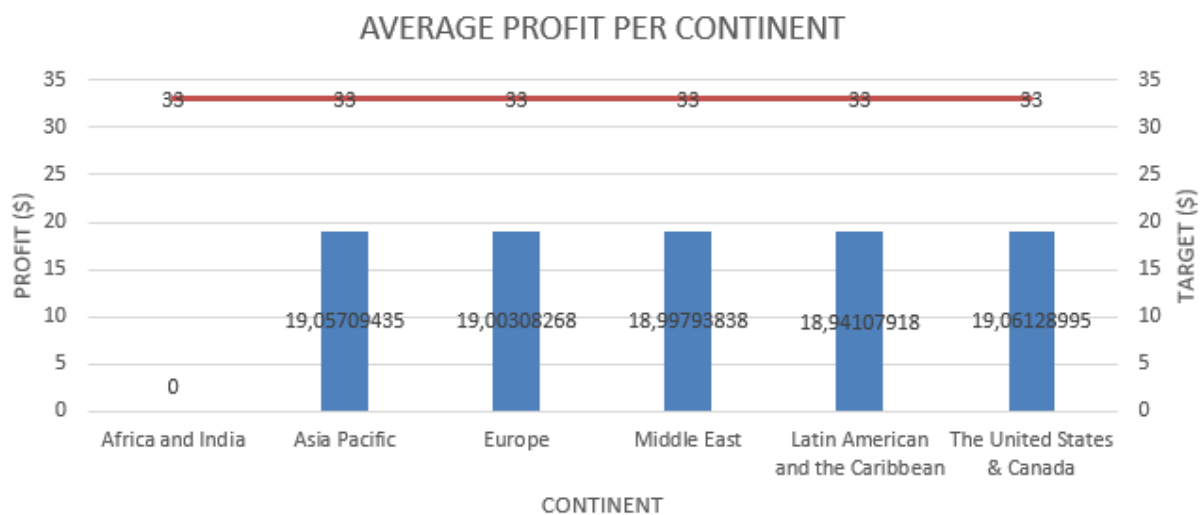


Gráfico de Barras sobre o *Lead Time* Médio por Mercado para um Dia

Number of orders	Continent	Average lead time	Target
2877	Africa and India	0	1,65
	Asia Pacific	1,302762373	1,65
	Europe	1,302986648	1,65
	Middle East	1,304785561	1,65
	Latin American and the Caribbean	1,304449982	1,65
	The United States & Canada	1,303301788	1,65

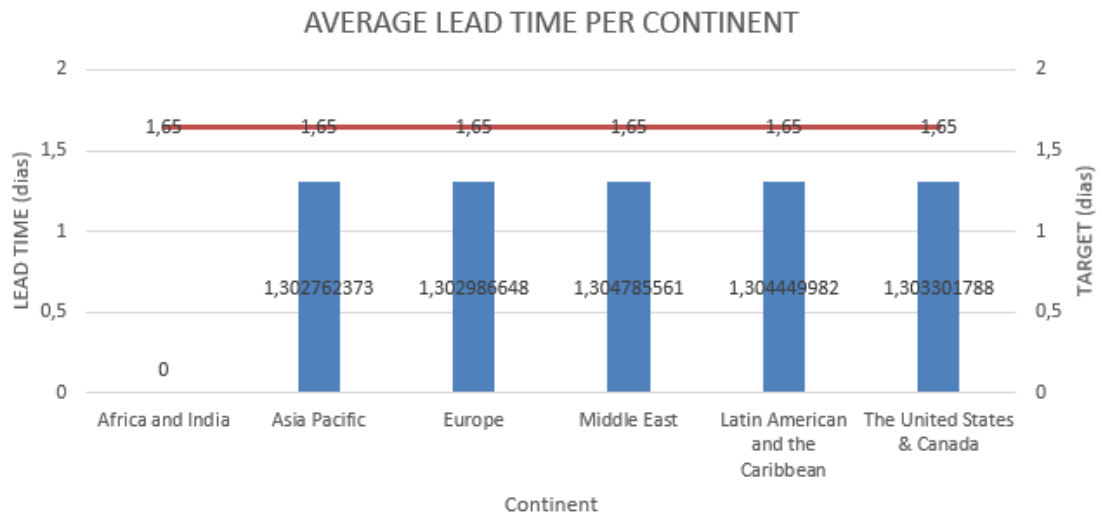


Gráfico de Barras sobre as Vendas Médias *Online* por Estação para uma Semana

Season	Online sales average	Target
AW09	3,666666667	23,1
AW10	11,78608425	23,1
AW11	12,080268	23,1
AW13	12,08151237	23,1
AW14	12,09885295	23,1
AW15	12,09076154	23,1
AW16	12,04971669	23,1
AW17	12,05779832	23,1
AW18	12,08223246	23,1
SS10	33	23,1
SS11	12,11223154	23,1
SS12	11,86857908	23,1
SS13	12,12718387	23,1
SS14	12,07285298	23,1
SS15	12,02297302	23,1
SS16	12,04943225	23,1
SS17	12,04983921	23,1
SS18	12,06054281	23,1
Archive	12,34163933	23,1
Carryover	12,03079657	23,1
Vintage	12,03815736	23,1

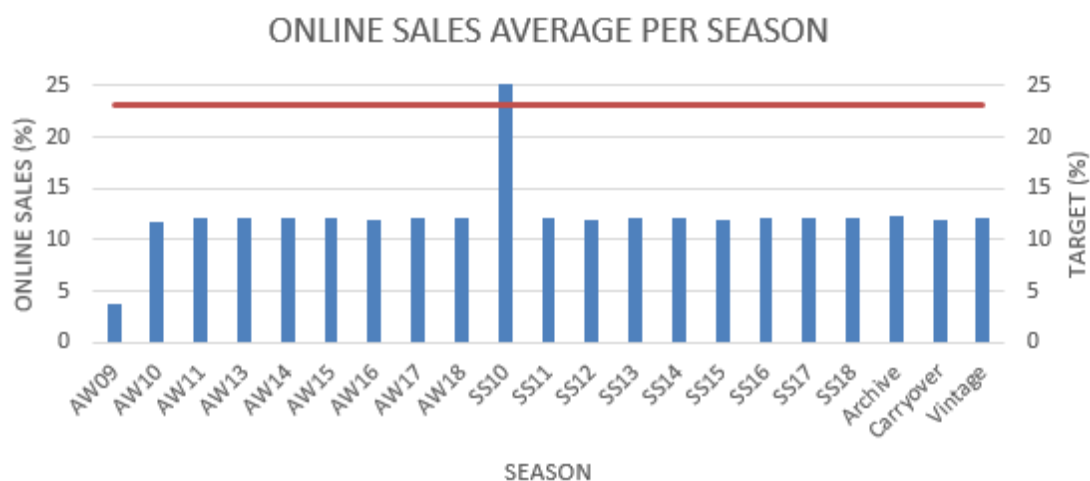
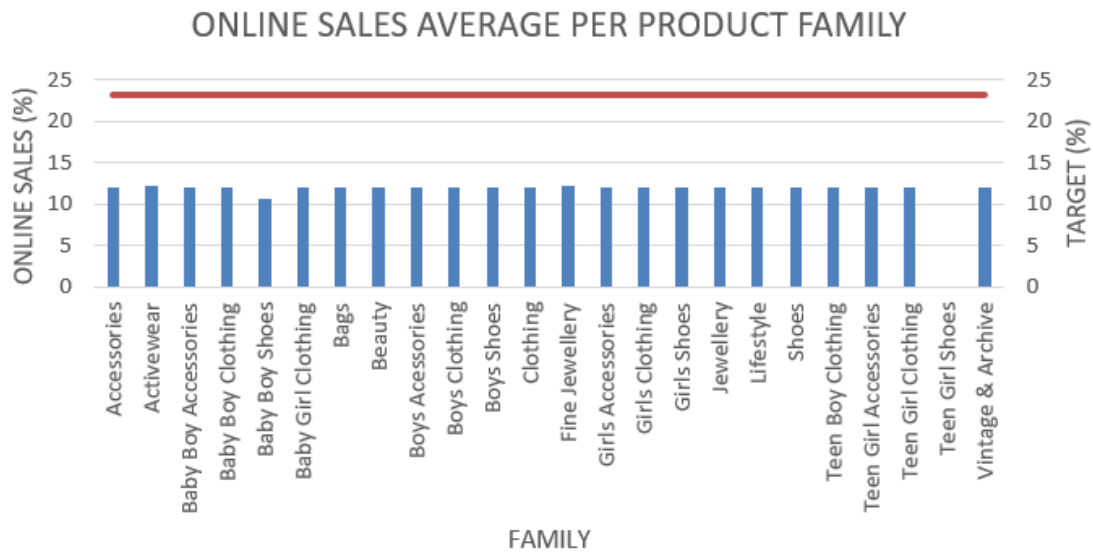


Gráfico de Barras sobre as Vendas Médias *Online* por Família de Produto para uma Semana

Family	Online sales average	Target
Accessories	12,04348333	23,1
Activewear	12,23014107	23,1
Baby Boy Accessories	12,09944449	23,1
Baby Boy Clothing	12,05327188	23,1
Baby Boy Shoes	10,68211079	23,1
Baby Girl Clothing	12,0753246	23,1
Bags	12,06199088	23,1
Beauty	12,05085538	23,1
Boys Aecessories	12,09622937	23,1
Boys Clothing	12,08124924	23,1
Boys Shoes	12,0476974	23,1
Clothing	12,05779832	23,1
Fine Jewellery	12,11735612	23,1
Girls Accessories	12,05199644	23,1

Girls Clothing	12,06154845	23,1
Girls Shoes	12,06049773	23,1
Jewellery	12,06071875	23,1
Lifestyle	12,05507931	23,1
Shoes	12,04971669	23,1
Teen Boy Clothing	12,06436857	23,1
Teen Girl Accessories	12,02481051	23,1
Teen Girl Clothing	12,03383817	23,1
Teen Girl Shoes	0	23,1
Vintage & Archive	12,03815736	23,1



Anexo G: Rede Neuronal

Código de Inicialização

```
class NN:
    def __init__(self, NI, NH, NO):

        self.ni = NI + 1
        self.nh = NH
        self.no = NO

        self.ai, self.ah, self.ao = [], [], []
        self.ai = [1.0]*self.ni
        self.ah = [1.0]*self.nh
        self.ao = [1.0]*self.no

        self.wi = makeMatrix (self.ni, self.nh)
        self.wo = makeMatrix (self.nh, self.no)

        randomizeMatrix ( self.wi, 0, 1)
        randomizeMatrix ( self.wo, 0, 1)

        self.ci = makeMatrix (self.ni, self.nh)
        self.co = makeMatrix (self.nh, self.no)
```

Código de Propagação Direta

```
def runNN (self, inputs):
    if len(inputs) != self.ni-1:
        print('incorrect number of inputs')

    for i in range(self.ni-1):
        self.ai[i] = inputs[i]

    for j in range(self.nh):
        sum = 0.0
        for i in range(self.ni):
            sum +=( self.ai[i] * self.wi[i][j] )
        self.ah[j] = sigmoid (sum)

    for k in range(self.no):
        sum = 0.0
        for j in range(self.nh):
            sum +=( self.ah[j] * self.wo[j][k] )
        self.ao[k] = sigmoid (sum)

    return self.ao
```


Código de Retropropagação

```
def backPropagate (self, targets, N, M):

    output_deltas = [0.0] * self.no
    for k in range(self.no):
        error = targets[k] - self.ao[k]
        output_deltas[k] = error * dsigmoid(self.ao[k])

    for j in range(self.nh):
        for k in range(self.no):

            change = output_deltas[k] * self.ah[j]
            self.wo[j][k] += N*change + M*self.co[j][k]
            self.co[j][k] = change

    hidden_deltas = [0.0] * self.nh
    for j in range(self.nh):
        error = 0.0
        for k in range(self.no):
            error += output_deltas[k] * self.wo[j][k]
        hidden_deltas[j] = error * dsigmoid(self.ah[j])

    for i in range (self.ni):
        for j in range (self.nh):
            change = hidden_deltas[j] * self.ai[i]
            self.wi[i][j] += N*change + M*self.ci[i][j]
            self.ci[i][j] = change

    error = 0.0
    for k in range(len(targets)):
        error = error + ( 0.5 * (targets[k]-self.ao[k])**2 )
    return error
```

Principais Funções

```
def weights(self):
    print('Input weights:')
    for i in range(self.ni):
        print(self.wi[i])
    print
    print('Output weights:')
    for j in range(self.nh):
        print(self.wo[j])
    print('')

def test(self, patterns):
    for p in patterns:
        inputs = p[0]
        print('Inputs:', p[0], '-->', 'Outputs:', self.runNN(inputs), '\tTarget', p[1])

def train (self, patterns, max_iterations = 2878, N=0.3, M=0.4):
    for i in range(max_iterations):
        for p in patterns:
            inputs = p[0]
            targets = p[1]
            self.runNN(inputs)
            error = self.backPropagate(targets, N, M)
        if i % 50 == 0:
            print('Combined error', error)

    self.test(patterns)

def sigmoid (x):
    return math.tanh(x)

def dsigmoid (y):
    return 1 - y**2

def makeMatrix (I, J, fill=0.0):
    m = []
    for i in range(I):
        m.append([fill]*J)
    return m

def randomizeMatrix (matrix, a, b):
    for i in range (len (matrix)):
        for j in range (len (matrix[0])):
            matrix[i][j] = random.uniform(a,b)
```

```

a = 0
b = 0
c = 0
d = 0
list = [[[0 for x in range(3)], [0 for x in range(3)]] for j in range(len(resultados))]
for x in range(len(list)):
    list[a][b][c] = resultados['Profit'][d]
    c = c+1
    list[a][b][c] = resultados['Partners'][d]
    c = c+1
    list[a][b][c] = resultados['Customers'][d]
    b = b+1
    c = 0
    list[a][b][c] = resultados['TargProfit'][0]
    c = c+1
    list[a][b][c] = resultados['TargPartners'][0]
    c = c+1
    list[a][b][c] = resultados['TargCustomers'][0]
    d = d+1
    a = a+1
    b = 0
    c = 0

def main ():
    pat = list
    myNN = NN (3, 1, 3)
    myNN.train(pat)
    myNN.weights()

if __name__ == "__main__":
    main()

```


Anexo H: Resultados da Rede Neuronal

Excerto de Resultados da Rede Neuronal

```
Inputs: [0.1847143236904076, 0.25, 0.1827676240208878] --> Outputs: [0.06929219300000004, 0.700000000000000002, 0.093994778000000007] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0709065445989101, 0.85, 0.09921671018276762] --> Outputs: [0.06929219300000002, 0.700000000000000001, 0.093994778000000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.05112703294606737, 0.0, 0.1958822454308094] --> Outputs: [0.06929219300000002, 0.700000000000000001, 0.093994778000000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.08905019101636087, 0.0, 0.19321144882506528] --> Outputs: [0.06929219300000002, 0.700000000000000001, 0.093994778000000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0323220283213093, 0.4285714285714285, 0.07832898172323761] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.093994778] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.3777671480793094, 0.5, 0.2219321144825065] --> Outputs: [0.06929219300000009, 0.70000000000000004, 0.09399477800000013] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.01966961626312294, 0.0, 0.05483028720626632] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.04688318560810732, 1.0, 0.09399477806788513] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0292273830077679, 0.0, 0.04438642297650131] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0335962480248278, 0.0, 0.04177545691906006] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0290951278851336, 0.5, 0.04699738903394257] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0728291069691521, 0.5, 0.1201044386422977] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0123231247580281, 0.1, 0.0287206266318538] --> Outputs: [0.06929219299999996, 0.6999999999999997, 0.09399477799999997] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0142132030106591, 1.0, 0.14621409921671] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0125245676066756, 1.0, 0.107049608355091] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000003] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0234466227466762, 0.0, 0.0678851174934726] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.0213213840842522, 0.75, 0.0234986945169713] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.02134000720407016, 0.0, 0.06266318537859009] --> Outputs: [0.06929219299999997, 0.6999999999999998, 0.09399477799999999] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.1656001783423487, 0.0, 0.1723237597911227] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000004] Target [0.069292193, 0.7, 0.093994778]
Inputs: [0.01094008379864407, 1.0, 0.1723237597911227] --> Outputs: [0.06929219300000002, 0.70000000000000001, 0.09399477800000004] Target [0.069292193, 0.7, 0.093994778]
```

Input weights:

[4.276844822450413e-15]

[5.104749519938111e-16]

[3.794836206994675e-15]

[0.8764219897892694]

Output weights:

[0.09849732718658669, 1.2308729411829749, 0.13379234433795315]

Anexo I: Resultados Após Validação

Gráfico de Barras sobre o Lucro Médio por Mercado para um Dia

Number of orders	Continent	Average profit	Target
2877	Africa and India	0	9,85
	Asia Pacific	6,831314066	9,85
	Europe	6,808939173	9,85
	Middle East	6,816271851	9,85
	Latin American and the Caribbean	6,798953436	9,85
	The United States & Canada	6,829693769	9,85

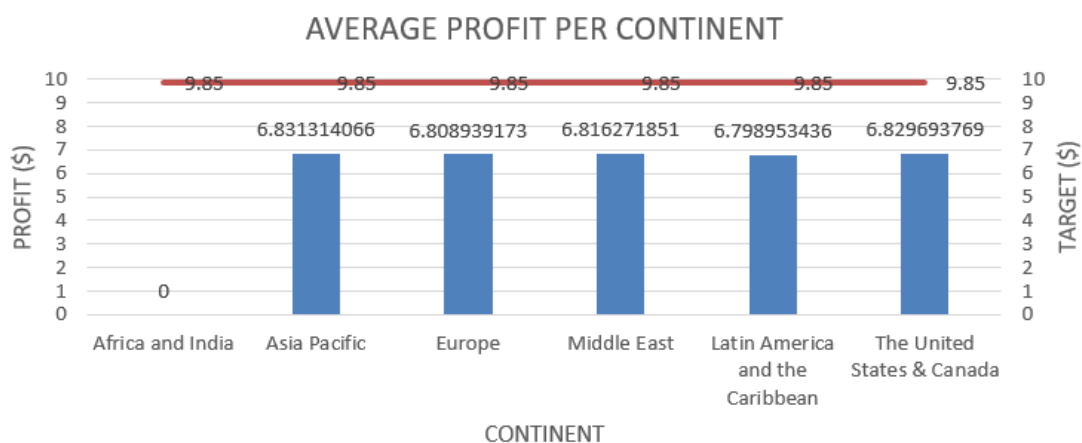


Gráfico de Barras sobre o *Lead Time* Médio por Mercado para um Dia

Number of orders	Continent	Average lead time	Target
2877	Africa and India	0	0,669
	Asia Pacific	0,584326986	0,669
	Europe	0,583821464	0,669
	Middle East	0,584782215	0,669
	Latin American and the Caribbean	0,584790505	0,669
	The United States & Canada	0,583920501	0,669

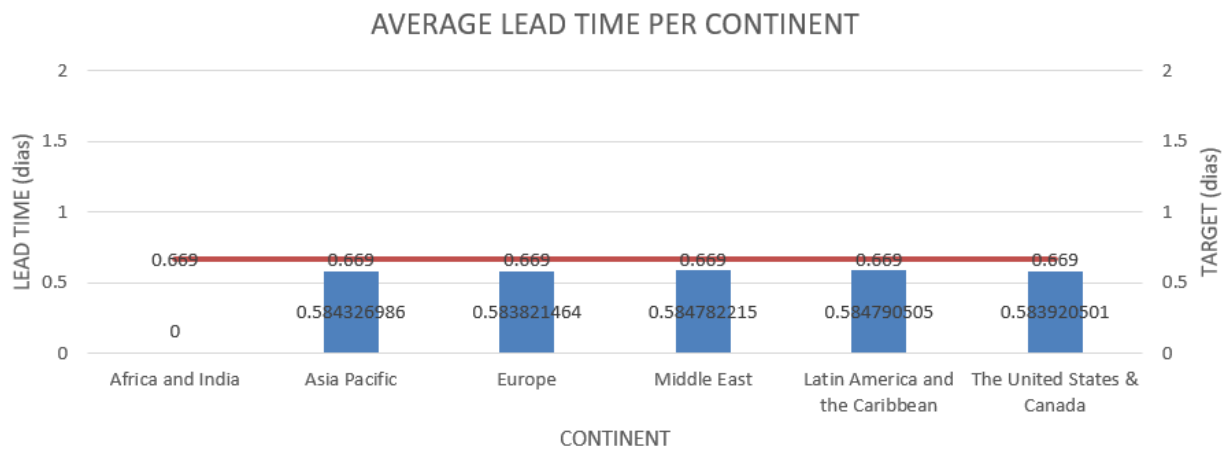


Gráfico de Barras sobre as Vendas Médias *Online* por Estação para uma Semana

Season	Online sales average	Target
AW09	13.67666667	86.163
AW10	43.96209425	86.163
AW11	45.05939963	86.163
AW13	45.06404113	86.163
AW14	45.12872149	86.163
AW15	45.09854053	86.163
AW16	44.94544324	86.163
AW17	44.97558772	86.163
AW18	45.06672707	86.163
SS10	123.09	86.163
SS11	45.17862365	86.163
SS12	44.26979996	86.163
SS13	45.23439582	86.163
SS14	45.03174161	86.163
SS15	44.84568936	86.163
SS16	44.94438229	86.163
SS17	44.94590026	86.163
SS18	44.98582469	86.163
Archive	46.03431469	86.163
Carryover	44.87487121	86.163
Vintage	44.90232695	86.163

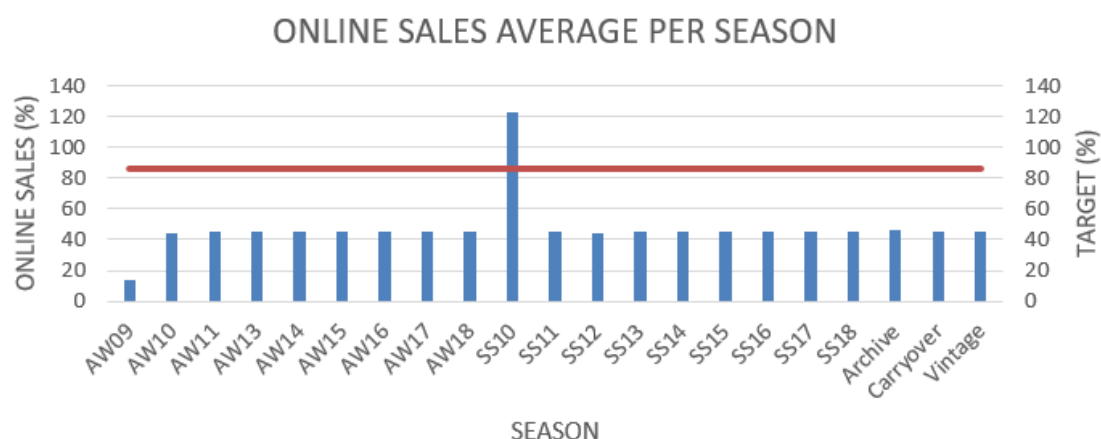


Gráfico de Barras sobre as Vendas Médias *Online* por Família de Produto para uma Semana

Family	Online sales average	Target
Accessories	44.92219282	86.163
Activewear	45.61842618	86.163
Baby Boy Accessories	45.13092794	86.163
Baby Boy Clothing	44.95870411	86.163
Baby Boy Shoes	39.84427324	86.163
Baby Girl Clothing	45.04096076	86.163
Bags	44.99122597	86.163
Beauty	44.94969058	86.163
Boys Aecessories	45.11893555	86.163
Boys Clothing	45.06305965	86.163
Boys Shoes	44.93791131	86.163
Clothing	44.97558772	86.163
Fine Jewellery	45.19773831	86.163
Girls Accessories	44.9539467	86.163

Girls Clothing	44.98957571	86.163
Girls Shoes	44.98565654	86.163
Jewellery	44.98648092	86.163
Lifestyle	44.96544582	86.163
Shoes	44.94544324	86.163
Teen Boy Clothing	45.00009476	86.163
Teen Girl Accessories	44.85254321	86.163
Teen Girl Clothing	44.88621636	86.163
Teen Girl Shoes	0	86.163
Vintage & Archive	44.90232695	86.163

